UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2020-69
The Faculty of Science, Technology and Medicine
The Faculty of Law, Economics and Finance
The Faculty of Humanities, Education and Social Sciences

DISSERTATION

Presented on 09/11/2020 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Jérémie DAUPHIN
Born on 8 March 1992 in Nice (France)

ARGUMENT ACCEPTANCE AND COMMITMENT IN FORMAL ARGUMENTATION

To go wrong in one’s own way is better than to go right in someone else’s.
Fyodor Dostoevsky

ACKNOWLEDGEMENTS

First and foremost I would like to thank my supervisor and mentor, the head of the ICR research group, Prof. Leendert van der Torre. I am extremely grateful not only for all the support, feedback and inspiration he has given me, but also for all the trust and confidence he has put in me throughout this venture, providing me with the opportunities and room to learn and grow.

I would also like to thank Dr. Marcos Cramer, my advisor who gave me so much support, in particular in the beginning, which allowed me to get a smooth start towards this thesis. I will always remember the interesting scientific discussions we've had, sometimes until late at night. Marcos' contract at the University of Luxembourg ended before mine, and so it was a pleasure to go visit him and his new research group in Dresden.

I am also grateful to Prof. Ken Satoh, head of the research group I visited for an extended research stay. I thank him for providing me with new insights and a different point of view on the problems I was working on. I also thank him for the warm welcome in Tokyo and for showing me various aspects of the Japanese culture that I would not have discovered otherwise.

Prof. Beishui Liao has closely followed my work and provided continuous feedback on my research, for which I am grateful. He was also a wonderful host during the CLAR2018 conference, and I am sure would have been a wonderful host again for the CLAR2020 conference as well had the circumstances been different.

The ICR group has always had a wonderful atmosphere, extremely supportive and kind-hearted, and so I would like to thank all of its members, present and past. In particular I would like to mention Shohreh Haddadan with whom I shared my office for most of my time as a PhD student, I wish her good luck for her thesis as well!

I would like to thank my family for the continuous support, in particular during the hard times of the pandemic when my moral had taken a hit. I also thank all my friends, in particular *the gang*: you guys know who you are, stay awesome.

Finally, I would like to thank all the interesting people I met during this journey. To Tiago Oliveira and the Tokyo group, I'm sure we'll meet again. I thank the folks from the ESSLLI summer school, every edition was full of interesting courses, social activities and amazing people: Marianela Morales, Nenad Savić, Maya Olszewski and many more. I have also met great people at the SoAIR spring school: Alexandre Angleraud, Edgar Handy, Derrick Odonkor and many more. I would additionally like to thank the formal argumentation scientific community for being such a supportive and friendly community.

Table of Contents

List of Tables

List of Figures

SUMMARY

Formal argumentation is an area of symbolic reasoning which mimics the way people argue with each other, providing arguments to back their claims and counter-arguments to attack opposing claims. These are then represented in an *argumentation framework*. The *acceptability* of such arguments is then evaluated by abstracting away their internal content, so that the process only considers the relation of *attack* between them. This then results in *extensions*, representing reasonable, rational stances on the acceptability status of the arguments. While rationality constraints are set within an extension, different extensions are usually not compatible as they can represent opposing views on the same situation. In this thesis, we focus on the aspect of moving from acceptability to *commitment* towards one such extension. We present a framework for gradual commitment and study its properties in different settings.

We later investigate how this framework allows for the combination of different *semantics*, the functions which, given an argumentation framework, return a set of extensions. This combination is done by allowing one to switch semantics in the middle of the commitment process. We show that this combination allows one to recover an existing semantics from two other existing ones and study how this method allows us to define new semantics.

The acceptability of arguments is derived from the attack relation, but other relations have been introduced for formal argumentation, such as positive relations of support and explanations, and more complex relations such as joint attacks. We then present a framework which allows for not only attacks, but also necessary and deductive support, explanations and incompatibilities between any set of elements. This enriched framework, which we call *extended explanatory argumentation framework*, is then equipped with both labelling semantics, where instead of extensions one has labeling functions which assign to each element of the framework one of three acceptability statuses: *in*, *out* or *undec*. We also define extension-based semantics via a translation from an enriched framework to a simpler framework using a *flattening* approach, and show that there is a correspondence between the two approaches.

We then propose a structured argumentation framework, allowing for the construction of arguments from a knowledge base and a set of rules. This framework has the particularities that it allows for hypothetical reasoning in the construction of arguments, and explanations of other arguments and explananda, which are scientific observations the arguments attempt to explain.

Chapter 1

Introduction

1.1 Artificial Intelligence

Artificially intelligent machines are omnipresent in our daily lives, from smart TVs to lawn-mowing robots, but also at the professional level, with autopilot for airplanes [1] and surgical robots with near perfect precision [2].

The field is split in two main methodologies. One the one hand, there is the *machine learning* approach, where large samples of data are fed into a learning algorithm, allowing the machine to identify the most relevant features of the data with respect to the task and how they affect it. For example, by feeding tens of thousands of images from ten different classes, labelled with their corresponding class (one of them being dog), a well set up machine is then able to identify from new images in which of the ten classes it belongs (e.g. whether it is a picture of a dog or not) [3]. This approach takes advantages of the computing power of machines, but unfortunately often lacks transparency. Once all the data has been fed through the machine learning algorithm, even experts can have a hard time understanding exactly how the machine interprets the new data and how it justifies its output.

On the other hand, there is the *symbolic reasoning* approach, where instead the aim is to teach the machine to reason in a manner more similar to how humans think, using for instance logical rules of deduction and a knowledge base. One example would be specifying to a machine that birds fly, and that the animal named Tweety is a bird. The machine is then able to additionally deduce that Tweety flies. This results in general in more transparency, since the machine can now mention these two facts as justification for its conclusion.

This thesis focuses on a particular field of symbolic reasoning, namely the field of *formal argumentation*, popularised by the work of Dung [4]. Here, the aim is to mimic the kind of reasoning which occurs when humans argue with each other, putting forward arguments to support their claims and counter-arguments to attack arguments they do not agree with.

The field is further sub-divided into two main approaches. A higher-level approach of *abstract argumentation* focuses on the reasoning aspects, investigating the *acceptability* of arguments based solely on their relation with other arguments. Note that while the field of logic is mostly devoted to evaluating the *truth* of propositions, the field of formal argumentation focuses on their *acceptability*. For example, two arguments with opposite conclusions might both be acceptable to the same degree, although not jointly. So even though one might not be able to reasonably accept both arguments at the same time, accepting either of them separately would be perfectly rational. In this sense, formal argumentation is similar to *answer set programming* [5] which also focuses on providing multiple possible answer sets which each solve the same given problem but might not be compatible with each other. In Chapter 3, we investigate the process of *committing* towards one such set of acceptable arguments.

On the other hand, the sub-field of *structured argumentation* focuses on the *construction* of arguments from various structures, for example a knowledge base with a set of potentially defeasible rules. Various frameworks have been devised for structured argumentation, most notably ASPIC+ [6], Assumption-Based Argumentation (ABA) [7] and Defeasible Logic Programming (DeLP) [8]. In Chapter 6 we investigate an extension of the ASPIC+ framework which allows for explanations and hypothetical reasoning.

1.2 Argumentation

In abstract argumentation, a semantics is a function which takes an abstract argumentation framework and returns a set of *extensions*, where an extension is a set of jointly acceptable arguments. Different semantics have been proposed in the literature [9], being more or less appropriate depending on the application and context. Most of these semantics return multiple extensions [10]. This gives rise to two crucial questions: On the one hand, from all the semantics, which one should be applied? And on the other hand, how does one choose from multiple extensions?

One avenue to helping with the first question is the principle-based approach for abstract argumentation [11, 12, 10]. By defining guiding principles that may or may not be desirable depending on the context, one can select a semantics which behaves best for the application of interest.

For the second question, two approaches exist: one can define higher-level concepts of acceptability based on whether an argument is in no extension, at least one, or all. For example, an argument is said to be *strongly accepted* iff it is in all extensions for a given semantics. The issue is that in this process, many of the desirable properties defined in the aforementioned mentioned principles are lost. For instance, admissible sets of arguments are sets of arguments which attack any argument attacking an element of the set, without having any attack within the set. The preferred semantics returns $\subseteq$-maximal admissible sets of arguments. Interestingly, the set of strongly accepted arguments with respect to preferred semantics is not necessarily admissible. This issue is avoided in the second approach, to select a single extension from the proposed ones. This further increases the importance of methods for making this selection.

Many different methods for selecting an extension already exist, so we propose instead to study the selection process itself. By refining the extraction of sets of acceptable arguments from a framework into smaller steps, one can construct a graph which represents different levels of commitments towards some extensions, until a single one has been elected. We therefore introduce the notion of *commitment graphs* for choosing between multiple extensions of an argumentation framework in a given semantics. The edges of a commitment graph represent specific commitment steps that bring one closer to the final commitment where no alternative subsist and only a single extension remains. Following the distinction that Dung [4] introduced between abstract argumentation frameworks and structured instantiations thereof, we propose to make a distinction between *abstract commitment graphs* and *concrete commitment graphs*, allowing us to distinguish features which arise from the structure alone of such a commitment graph, and study these separately. Note also that our formal notion of concrete commitment graphs forces a specific kind of instantiations, of which variations could be studied in the future. This separation between abstract and concrete commitment graphs further allows for transferability of some results.

In the formal argumentation literature, the *labeling approach* [13] is often used to determine the acceptability of arguments. Here, one derives labeling functions which assign to each argument one of three labels: *in*, *out* or *undec*. The arguments that are labeled *in* represent the arguments that are jointly acceptable, while the arguments that are *out* represent the ones that are defeated by those. The last label, *undec* (*undecided*), represents the cases where one cannot, or decides with proper justification, not to assign either of these two labels, because their situation allows for them to be rejected without the acceptance of a counter-argument, such as in dilemmas or paradoxical structures. One advantage of the labeling approach is that to verify that an argument is correctly labeled, one only needs to check the labels of its direct ancestors. This allows for a more local evaluation, which is still equivalent to other global approaches such as the extension-based approach, which states what properties a set of arguments must satisfy in order to be an extension. In the second chapter of this thesis, we propose one construction for concrete commitment graphs which relies on the concept of *partial extensions*. Here one also distinguishes three statuses for arguments: arguments which have been accepted, arguments which have been rejected, and lastly arguments on whose status one hasn't committed yet. We however draw a clear distinction between the *undec* label and our 'uncommitted' status, as we assign the rejected status to *undec* arguments, meaning one has committed not to accept the arguments in question. Our uncommitted status on the other hand simply represents an abstention of judgement on the argument, where it is still possible to either accept or reject the argument.

In Chapter 4 of this thesis, we refine the notion of partial extension into *epistemic labels*. Here, every arguments is assigned the set of potential labels *in*, *out* and *undec*. One then narrows it down until every argument only has a single possible label remaining, providing more intermediary steps than in the partial extension approach. This more granular construction is then used to combine different semantics, allowing for the systematic creation of new semantics based on existing ones.

Several methods for combining argumentation semantics have been studied. For example, in multi-sorted argumentation [14, 15, 16], one part of the framework can be evaluated according for example to the grounded semantics, whereas another part of the framework is evaluated according to the preferred semantics. Another approach manipulates directly the sets of extensions. For example, the grounded and preferred can be combined by simply returning both the grounded and preferred extensions. Both of these approaches have drawbacks. For multi-sorted argumentation, we need to specify explicitly which semantics must be applied to which part of the framework. For the direct combination method, the approach seems too coarse-grained and the number of ways to combine semantics seems relatively limited.

The refinement of the aforementioned process of creating extensions brings up another interesting research question. If the process of extracting a set of acceptable arguments from a framework is done in smaller steps, is it possible to change semantics mid-way? This would allow one to combine semantics in a novel way. One would therefore be able to start accepting arguments with a broad-minded point of view, but then restrict one's mindset after having committed to accepting some number of arguments. In particular, from the complete extensions, one can further refine them into the unique grounded extension and the maximal-minded preferred extensions. The remaining extensions which are neither grounded nor preferred lie somewhere in the middle of the spectrum. Would it be possible to retrieve those extensions by alternating between preferred and grounded commitment mindsets? Though the derivation of the complete semantics from the grounded and preferred semantics does not serve any practical purpose, it serves to show that our dynamic semantic framework has sufficient expressive power to combine abstract semantics.

Several families of semantics exist, mainly admissibility-based semantics and naive-based semantics. They are known to satisfy quite different sets of principles, which raises another interesting question: Would a meaningful combination of two semantics from different families be possible, and what kind of results would it yield? Note that recently naive-based semantics like stage semantics [17] and CF2 semantics [18] have received some attention, for example in the work of Gaggl and Dvořák [19], who define a new semantics (*stage2*) that combines features of stage and CF2 semantics, and in the works of Cramer and Guillaume [20, 21], who performed empirical studies that showed that these naive-based semantics are better predictors of human argument acceptance than complete-based semantics like the grounded and preferred semantics. These cognitive studies have additionally sparked ideas for new semantics, such as the SCF2 semantics [22].

For argumentation frameworks without odd cycles, the stage semantics fully agrees with the preferred semantics. One difference between the preferred semantics and the stage semantics is that the stage semantics generally provides a way to select accepted arguments even when odd cycles are around, whereas the preferred semantics tends to mark as *undecided* all arguments that are in an odd cycle or attacked by an odd cycle. One difference between the preferred semantics and the complete semantics is that the complete semantics allows one to locally not make choices for some unattacked even cycles while making choices for other unattacked even cycles, whereas in the preferred semantics one has to make choices for all unattacked even cycles. This motivates the following research question: Is there a sensible semantics that allows one to locally make choices for some unattacked odd or even cycles while not making choices for other unattacked odd or even cycles? In Chapter 4, we construct such as semantics by combining the grounded and the stage semantics.

In Dung's abstract argumentation frameworks, the acceptability of arguments is evaluated based on the attack relation between arguments. However, various researchers have felt the need to extend abstract argumentation frameworks in order to model features of argumentation that cannot be directly modeled in abstract argumentation frameworks, e.g. by enriching them with recursive (higher-order) attacks [23], joint attacks [24], a support relation between arguments [25, 26], or explanatory features [27].

In Chapter 5, we incorporate multiple enrichments into a single framework, and allow these relations to occur not only between arguments, but also to originate and target sets of elements of any nature. For example, this framework allows for the representation of an argument attacking a set of two other arguments, so that one of them can be accepted, but not both together. One can then also have a set of two arguments necessarily support this attack, so that the attack is only active as long as both source arguments are accepted.

The semantics of this framework are defined through a labelling approach, and then also through a flattening approach, where the enriched framework is first translated to a simpler framework with less enrichments and existing semantics. This simpler framework is then evaluated and the extensions are translated into extensions for the original enriched framework.

Structured argumentation investigates the question of how arguments are generated in the first place. In Chapter 6, we study a structured argumentation framework, called ASPIC-END, which extends the widely used ASPIC+ framework [6]. In this extended framework, arguments can be constructed by introducing facts from the knowledge base, or applying reasoning rules to the conclusions of existing arguments, as in the ASPIC+ framework. Additionally, in ASPIC-END arguments can be constructed with hypothetical reasoning structures such as reasoning by contradiction. Here, one may introduce an assumption which can be absent from the knowledge base, but upon reaching a logical contradiction from this assumption, may then conclude that its negation holds. ASPIC-END also allows for the generation of explananda and an explanatory relation.

The research questions we address in this thesis therefore are the following:

- What can be gained by refining the process of selecting an extension from a set of extensions?
- How can this refined structure allow us to combine different semantics?
- How can we evaluate acceptability in a framework combining many enrichments?
- How can arguments be constructed for an enriched framework with more than just an attack relation?

The layout of this thesis is as follows: we first provide an overview of existing notions from the literature that this thesis builds upon in Chapter 2. In Chapter 3 we define and investigate the commitment graphs and their properties. In Chapter 4 we study how this structure allows us for the combining of different argumentation semantics, and show that this combination method allows one to retrieve complete from preferred and grounded. In Chapter 5 we investigate enrichments of Dung's abstract framework and the flattening methodology which allows one to represent some enrichments in terms of basic elements, and show how these enrichments allow us to model in a much more faithful fashion scientific debates from the philosophical literature. In Chapter 6 we investigate how some of these enrichments translate to a structured argumentation setting, and provide a case study of the structured framework for debates from the philosophical and history of mathematics literature. We discuss future work in each chapter, but also in Chapter 7 when preliminary results are already available. Finally, we conclude in Chapter 8.

Chapter 2

Preliminaries

In this chapter, we provide definitions for existing notions from the literature we make use of in this thesis. We also provide Tables 2.1 and 2.2 listing the main formal notions discussed in this thesis, along with the corresponding notation and a reference to the definition introducing the notion.

Name	Naming convention	Components	Reference
Argumentation Framework (AF)	F	$\langle \mathcal{A}, \rightarrow \rangle$	Def. 2.1.1
Semantics	σ	$\sigma(F) = E$	Def. 2.1.3
Extension	E	$E \subseteq \mathcal{A}$	Def. 2.1.3
Labeling	Lab	$\mathit{Lab} : \mathcal{A} \rightarrow \{\mathit{in}, \mathit{out}, \mathit{undec}\}$	Def. 2.1.5
Strongly Connected Component (SCC)	S	$S \in SCCS_F$	Def. 2.1.8
Explanatory AF (EAF)	F	$\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$	Def. 2.1.10
AF with Recursive Attacks (AFRA)	F	$\langle \mathcal{A}, \rightarrow \rangle$	Def. 2.1.15
Bipolar AF (BAF)	F	$\langle \mathcal{A}, \rightarrow, \Rightarrow_d \rangle$	Def. 2.1.19
AF with Necessities (AFN)	F	$\langle \mathcal{A}, \rightarrow, \Rightarrow_n \rangle$	Def. 2.1.21
Higher level AF	F	$\langle S, S^0, \rightarrow \rangle$	Def. 2.1.23
Argumentation System	AS	$(\mathcal{L}, \mathcal{R}, n)$	Def. 2.2.1
Knowledge Base	$\mathcal{K}$	$\mathcal{K} \subseteq \mathcal{L}$	Def. 2.2.2
Argumentation Theory	AT	$(AS, \mathcal{K})$	Def. 2.2.3
ASPIC+ Argument	A	Many	Def. 2.2.4
Abstract Commitment Graph	G	(V, E, L)	Def. 3.2.2
Partial Extension	Γ	$\Gamma \subseteq \{+, -\} \times \mathcal{A}$	Def. 3.3.1
Concrete Commitment Graph (CCG)	G	(V, E, L)	Def. 3.3.5
Intersection-based Commitment Graph (ICG)	$\texttt{icg}_\sigma(F)$	(V, E, L)	Def. 3.3.6
Most Exhaustive Update	$\mathit{meu}(F, \sigma)$	(V, E, L)	Def. 3.3.7
Most Fine-Grained Commitment Graph	$\mathit{mfg}_\sigma(F)$	(V, E, L)	Def. 3.3.11
SCC-Directional Commitment Graph	$\mathit{sdcg}_\sigma(F)$	(V, E, L)	Def. 3.4.3
Distance-Based Commitment Graph	DBCG	(V, E, L)	Def. 3.5.3
ACG-Equivalence	$\simeq^a_g$	$F \simeq^a_g F'$	Def. 3.6.1
ACG-Summary	G	(V, E, L)	Def. 3.6.4
CCG-Equivalence	$\simeq^c_g$	$F \simeq^c_g F'$	Def. 3.6.6
CCG-Summary	G	(V, E, L)	Def. 3.6.6

Table 2.1: The main formal notions discussed in this thesis.

Name	Naming convention	Components	Reference
Labeled Argumentation Framework (LAF)	F	$(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$	Def. 4.3.1
Epistemic Labeling	EpLab	$\mathit{EpLab} : \mathcal{A} \rightarrow \mathcal{P}(\mathbb{L}) \setminus \{\emptyset\}$	Def. 4.3.1
Update Relation	upd	$\mathit{upd} \subseteq \mathbb{F} \times \mathbb{F}$	Def. 4.3.5
Most Fine-grained Update	mfg_σ	$\mathit{mfg}_\sigma \subseteq \mathbb{F} \times \mathbb{F}$	Def. 4.3.10
Step Grounded Update	$\mathit{step\_grnd}$	$\mathit{step\_grnd} \subseteq \mathbb{F} \times \mathbb{F}$	Def. 4.4.2
Step Preferred Update	$\mathit{step\_pref}$	$\mathit{step\_pref} \subseteq \mathbb{F} \times \mathbb{F}$	Def. 4.4.4
Update Merge	$\uplus$	$\mathit{upd}_1 \uplus \mathit{upd}_2$	Def. 4.5.1
EAF Labeling	Lab	$(\mathit{Lab}_\mathcal{A}, \mathit{Lab}_{\dashrightarrow})$	Def. 5.2.1
AC-labeling	Lab	$(\mathit{Lab}_\mathcal{A}, \mathit{Lab}_{\dashrightarrow})$	Def. 5.2.4
EC-labeling	Lab	$(\mathit{Lab}_\mathcal{A}, \mathit{Lab}_{\dashrightarrow})$	Def. 5.2.8
Extended EAF (EEAF)	F	$\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$	Def. 5.4.1
EEAF Labeling	Lab	$(\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$	Def. 5.4.5
ASPIC-END Argumentation Theory	Σ	$(\mathcal{L}, \mathcal{R}, n, <)$	Def. 6.3.1
ASPIC-END Argument	A	Many	Def. 6.3.2

Table 2.2: The main formal notions discussed in this thesis, continued.

2.1 Abstract Argumentation

Abstract argumentation is a form of symbolic reasoning, based on the way people argue back and forth when debating an issue. Arguments are represented together with a relation of conflict between them. This relation is directed, and called *attack*, so that when presented with a counter-argument b to an argument a, we say that b attacks a. Contrary to most logical formalisms, the goal here is not to determine what is true, but to evaluate which arguments are *acceptable*. For example, two people might be arguing about why the football team they support is the best. Most often, neither of them are wrong, and while the two points of view are in conflict, they can both be justified in a reasonable fashion, and we will therefore say that they are *acceptable*. In abstract argumentation, the focus is on the arguments and the relation between them. Any other information is abstracted away from the system, allowing for the evaluation of acceptability to be done irrespective of factors such as the source of arguments, their names or even their internal content.

2.1.1 Dung's Argumentation Frameworks

We define the required notions from abstract argumentation as introduced by Dung [4] and as explained in its current state-of-the-art form by Baroni et al. [9].

We start by defining the fundamental notion of *argumentation frameworks* and the auxiliary notions of *$\rightarrow$-paths* and *odd $\rightarrow$-cycles*.

Definition 2.1.1 (Argumentation framework (AF))**.** An *argumentation framework (AF)* $F = \langle \mathcal{A}, \rightarrow \rangle$ is a finite directed graph in which the set $\mathcal{A}$ of vertices is considered to represent arguments and the set $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ of edges is considered to represent the attack relation between arguments, i.e. the relation between a counterargument and the argument that it counters.

Definition 2.1.2 ($\rightarrow$-path and odd $\rightarrow$-cycle)**.** An *$\rightarrow$-path* is a sequence $\langle a_0, \ldots, a_n \rangle$ of arguments where $(a_i, a_{i+1}) \in \rightarrow$ for $0 \leq i < n$ and where $a_j \neq a_k$ for $0 \leq j < k \leq n$ with either $j \neq 0$ or $k \neq n$. An *odd $\rightarrow$-cycle* is an $\rightarrow$-path $\langle a_0, \ldots, a_n \rangle$ where $a_0 = a_n$ and n is odd.

Given an argumentation framework, we want to choose sets of arguments for which it is rational and coherent to accept them together. A set of arguments that may be accepted together is called an *extension*. Multiple *argumentation semantics* have been defined in the literature, i.e. multiple different ways of defining extensions given an argumentation framework. Before we consider specific argumentation semantics, we first give a formal definition of the notion of *argumentation semantics*:

Definition 2.1.3 (Argumentation semantics)**.** An *argumentation semantics* is a function σ that maps any AF $F = \langle \mathcal{A}, \rightarrow \rangle$ to a set $\sigma(F)$ of subsets of $\mathcal{A}$. The elements of $\sigma(F)$ are called σ *extensions* of F.

Note 1. We usually define an argumentation semantics σ by specifying criteria which a subset of $\mathcal{A}$ has to satisfy in order to be a σ extension of F.

In this thesis we consider various semantics:

Definition 2.1.4 (Main argumentation semantics)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF, and let $S \subseteq \mathcal{A}$. The set S is called *conflict-free* iff there are no arguments $b, c \in S$ such that b attacks c (i.e. such that $(b, c) \in \rightarrow$). Argument $a \in \mathcal{A}$ is *defended* by S iff for every $b \in \mathcal{A}$ such that b attacks a there exists $c \in S$ such that c attacks b. We say that S is *admissible* iff S is conflict-free and every argument in S is defended by S.

- S is a *complete extension* of F iff S is admissible and S contains all the arguments it defends.
- S is a *stable extension* of F iff S is admissible and S attacks all the arguments of $\mathcal{A} \setminus S$.
- S is the *grounded extension* of F iff S is a minimal with respect to set inclusion complete extension of F.
- S is a *preferred extension* of F iff S is a maximal with respect to set inclusion complete extension of F.
- S is a *semi-stable extension* of F iff it is a complete extension and $S \cup S^+$ is maximal with respect to set inclusion among complete extensions, i.e. there exists no complete extension S_1 such that $S \cup S^+ \subset S_1 \cup S_1^+$.
- S is a *stage extension* of F iff S is a conflict-free set and $S \cup S^+$ is maximal with respect to set inclusion, i.e. S is conflict-free, and there exists no conflict-free set S_1 such that $S \cup S^+ \subset S_1 \cup S_1^+$.
- S is a *naive extension* of F iff S is a maximal conflict-free set.

These argumentation semantics can also be characterized through the *labeling-based approach* [13]. A labeling of an AF $F = \langle \mathcal{A}, \rightarrow \rangle$ is a function which assign to each argument one of three labels: *in*, *out* or *undec*. The *in* label represents the case where an argument is accepted, the *out* label represents the case where an argument is rejected, and the *undec* represents the case where an argument cannot be accepted, yet there is no reason to fully reject it either.

We provide definitions for the 3-labeling semantics of argumentation frameworks as defined in [13]. Note that in Chapter 4 we make use of the multi-labeling approach, where a set of labels is assigned to each argument. Such a set represents the possible labels for a given argument. The standard approach corresponds to the case where arguments are given singleton sets as labels.

We define $\mathbb{L} = \{\textit{in}, \textit{out}, \textit{undec}\}$ to be the set of possible *labels*.

Definition 2.1.5 (3-labeling)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF. We say that any function L from $\mathcal{A}$ to $\mathbb{L}$ is a 3-labeling of F.

The 3-labeling approach makes use of the notions of *legal labels*.

Definition 2.1.6 (Legal Labeling)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF, $a \in \mathcal{A}$ an argument and L a 3-labeling of F. We say that a is:

- *legally in* with respect to L iff $L(a) = in$ and for all $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) = out$;
- *legally out* with respect to L iff $L(a) = out$ and for some $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) = in$;
- *legally undecided* with respect to L iff $L(a) = undec$ and for all $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) \neq in$ and for at least one such b, $L(b) = undec$.

If all arguments in $\mathcal{A}$ are legally labeled with respect to L, then we say that L is a *complete labeling* of F. A complete labeling with a minimal set of *in*-labeled arguments is called a *grounded labeling*. A complete labeling with a maximal set of *in*-labeled arguments is called a *preferred labeling*. A complete labeling without *undec*-labeled arguments is called a *stable labeling*. A complete labeling with a minimal set of *undec*-labeled arguments is called a *semi-stable labeling*.

For each extension there is a corresponding labeling that can be defined as follows:

Definition 2.1.7 (Labelings)**.** Let E be an extension of the AF $F = \langle \mathcal{A}, \rightarrow \rangle$ according to one of the argumentation semantics defined above. Then the 3-labeling *Lab* corresponding to E is defined as follows:

$$Lab(a) = \begin{cases} in & \text{if } a \in E \\ out & \text{if there is an argument } b \in E \text{ such that } b \text{ attacks } a \\ undec & \text{otherwise.} \end{cases}$$

We now provide preliminary notions of Strongly Connected Components (SCCs) from the literature [18]. These partition the graph into cells, such that there is an $\rightarrow$-path between any two arguments within a cell.

Definition 2.1.8 (Strongly Connected Component)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF. We say that $S \subseteq \mathcal{A}$ is a *strongly connected component (SCC)* iff S is a maximal set with respect to $\subseteq$ such that for all distinct $a, b \in S$, there is a path from a to b in $\rightarrow$. We denote the set of all SCCs in F by $SCCS_F$.

Definition 2.1.9 (*sccparents* and *sccanc*)**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and an SCC $S \in SCCS_F$, we define

$$sccparents_F(S) := \{P \in SCCS_F \mid P \neq S \text{ and } \exists a \in P, \exists b \in S, (a, b) \in \rightarrow\}$$

and recursively define

$$sccanc_F(S) := sccparents_F(S) \cup \bigcup_{P \in sccparents_F(S)} sccanc_F(P)$$

2.1.2 Explanatory Argumentation Frameworks

In scientific debates, the discussions are usually centered around some phenomenons or evidence. The different parties propose theories to explain them and argue about which one of these theories provides the best explanation for the phenomenons in question. In this kind of setting, arguments arise with two different kinds of goals: some arguments will try to establish *whether* some statements are true, while others will aim at determining *why* some of the phenomenons of interest occur.

With this idea in mind, D. Šešelja and C. Straßer have extended abstract argumentation framework with explanatory features [27]. In these frameworks, there are not only arguments but also explananda. These are scientific phenomenons of which, unlike arguments, the acceptability is not being questioned. These can be seen as observations about the world which could not be trivially predicted from our knowledge base, or where our current knowledge base would predict the opposite.

Definition 2.1.10 (Explanatory Argumentation Framework (EAF))**.** An *explanatory argumentation framework* (EAF) is a tuple $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$, where $\mathcal{A}$ is a set of arguments, $\mathcal{X}$ is a set of explananda, $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation, $\dashrightarrow \subseteq \mathcal{A} \times (\mathcal{A} \cup \mathcal{X})$ is an explanation relation from arguments to either explananda or other arguments, and $\sim \subseteq \mathcal{A} \times \mathcal{A}$ is a symmetric incompatibility relation.

Note that the incompatibility relation's purpose is to differentiate between the opposing theories, as scientists usually do not accept multiple explanations of a given phenomenon at the same time.

Definition 2.1.11. Let $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. A set of arguments $S \subseteq \mathcal{A}$ is said to be *conflict-free* if and only if there are no arguments $a, b \in S$ such that $(a, b) \in \rightarrow \cup \sim$.

Note that the definition of admissible sets still stands but with the revised definition of conflict-freeness.

Definition 2.1.12 (Explanation offered)**.** An *explanation* $X[e]$ for $e \in \mathcal{X}$ offered by a set of arguments S is a subset S' of S such that there exists a unique argument $a \in S'$ such that $a \dashrightarrow e$ and for all $a' \in S' \setminus a$, there exists a path in $\dashrightarrow$ from a' to a.

Example 2.1.1. Consider the EAF in Figure 2.1.1. Note that the incompatibility relation has been represented by a straight line with no arrow between a and b.

Here we have two explananda, e_1 and e_2. a explains both e_1 and e_2 while b explains only e_2. Consider the conflict-free set $\{a, d, f\}$. It contains two explanations for e_1, namely $X_1[e_1] = \{a\}$ and $X_2[e_1] = \{a, d\}$. Similarly, it offers two explanations for e_2. The conflict-free set $\{b, f\}$ however offers an explanation only for e_2.

For our goal of selecting the best theory from our model, we need a way to compare how much and how well a given set of arguments is able to explain. The notions of *explanatory power* and *explanatory depth* are thus borrowed from P. Thagard's theory of explanatory coherence [28], and adapted to the abstract argumentation setting. A theory can be broader by explaining more of the phenomenons of interest, and thus being more powerful. On the other hand, it can be deeper by being itself explained by other hypotheses, or in our

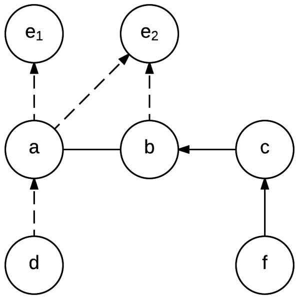


Figure 2.1: Example EAF1

case, other arguments. For example, in a legal setting, the hypothesis that a certain individual is guilty of a given crime is strengthened when it manages to explain the presence of additional evidence, but it can also be deepened when a motive which explains why the individual commited that crime is given.

Definition 2.1.13 (Explanatory power)**.** A set of arguments S_1 is *explanatorily more powerful* than a set of arguments S_2 ($S_1 >_p S_2$) if and only if the set of explananda for which S_1 offers an explanation is a strict super-set of the set of explananda for which S_2 offers an explanation.

A set of arguments S_1 is *explanatorily deeper* than a set of arguments S_2 ($S_1 >_d S_2$) if and only if for every explanation X_2 offered by S_2, there is an explanation X_1 offered by S_1 such that $X_2 \subseteq X_1$, and for at least one explanation X_1 offered by S_1, there is no explanation X_2 offered by S_2 such that $X_1 \subseteq X_2$.

In our previous example, we have that $\{a, d\} >_p \{b\}$ since $\{a, d\}$ offers an explanation for $\{e_1, e_2\}$ while $\{b\}$ only offers an explanation for $\{e_2\}$. Additionally, we have that $\{a, d\} >_d \{a\}$ and $\{a, d, f\} >_d \{a, f\}$.

Šešelja and Straßer [27] then propose two procedures for the selection of the best sets of arguments with respect to these notions. These have been revised as extensions by Cramer et al. [29], in order to be more in line with abstract argumentation extensions, while preserving their concepts.

Definition 2.1.14 (Satisfactory, Insightful, AC- and EC-extension)**.** Let $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $S \subseteq \mathcal{A}$ a set of arguments.

1. We say that S is *satisfactory* iff S is admissible and there is no $S' \subseteq \mathcal{A}$ such that $S' >_p S$ and S' is admissible.

2. We say that S is *insightful* iff S is satisfactory and there is no $S' \subseteq \mathcal{A}$ such that $S' >_d S$ and S' is satisfactory.

3. We say that S is an *argumentative core extension (AC-extension)* of Δ iff S is satisfactory and there is no $S' \supset S$ such that S' is satisfactory.

4. We say that S is an *explanatory core extension (EC-extension)* of Δ iff S is insightful and there is no $S' \subset S$ such that S' is insightful.

In our example, the only AC-extension is $\{a, d, f\}$, while the only EC-extension is $\{a, d\}$.

2.1.3 Argumentation Frameworks with Recursive Attacks

While EAFs add explanatory features to abstract argumentation frameworks, Baroni et al. [23] have developed an extension which enhances the expressive power of the attack relation. In their frameworks, they allow for attacks to target other attacks. This way, an argument may refute an attack relation between two other arguments without contesting the acceptability of any of them.

Definition 2.1.15 (Argumentation Framework with Recursive Attacks (AFRA))**.** An *Argumentation Framework with Recursive Attacks* (AFRA) is a pair $\langle \mathcal{A}, \rightarrow \rangle$ where $\mathcal{A}$ is a set of arguments and $\rightarrow \subseteq \mathcal{A} \times (\mathcal{A} \cup \rightarrow)$ is an attack relation from arguments to either arguments or attacks.

For a given attack $\alpha = (a, x) \in \rightarrow$, we say that the source of α is $src(\alpha) = a$ and its target is $trg(\alpha) = x$.

Now that attacks can be targeted, we need to extend our notions of acceptance to also include them.

Definition 2.1.16 (AFRA Defeat)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $\varphi \in \rightarrow, \psi \in (\mathcal{A} \cup \rightarrow)$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We say that φ *defeats* ψ iff either $\psi = trg(\varphi)$ or $src(\psi) = trg(\varphi)$.

Additionally, we say that S is *conflict-free* iff there do not exist $\varphi, \psi \in S$ such that φ defeats ψ.

The notions of defense and admissibility then follows with a similar idea as in standard abstract argumentation frameworks.

Definition 2.1.17 (AFRA admissible)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $\varphi \in (\mathcal{A} \cup \rightarrow)$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We say that S *defends* φ iff for every $\psi \in \rightarrow$ such that ψ defeats φ, there exists a $\delta \in S$ such that δ defeats ψ. We say that S is *admissible* iff S is conflict-free and defends its elements.

The complete semantics then follows with a similar definition as in classical abstract argumentation but using the adapted notions just defined.

Definition 2.1.18 (AFRA complete)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We say that S is a *complete extension* of F iff S is admissible and contains every $\varphi \in (\mathcal{A} \cup \rightarrow)$ it defends.

2.1.4 Support in Abstract Argumentation

While classical abstract argumentation revolves around attacks, there has been research on extending it with a positive relation of support between arguments. Many possible interpretations for this relation of support have been studied, in particular deductive support [26], necessary support [30] and evidential support [31]. We will first examine the formalism introduced by Cayrol and Lagasquie-Schiex called bipolar argumentation framework [25], as summarized by G. Boella et al. in [26], which focuses on support of the deductive kind.

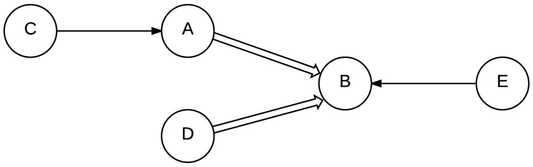


Figure 2.2: Example bipolar argumentation framework

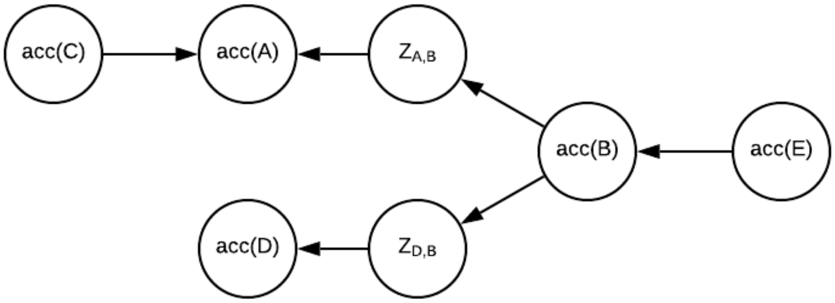


Figure 2.3: Flattened BAF from Figure 2.2

Definition 2.1.19 (Bipolar Argumentation Framework (BAF))**.** A *bipolar argumentation framework* (BAF) is a triple $\langle \mathcal{A}, \rightarrow, \Rightarrow_d \rangle$ where $\mathcal{A}$ is a set of arguments, $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation and $\Rightarrow_d \subseteq \mathcal{A} \times \mathcal{A}$ is a deductive support relation.

Boella et al. [26] treat support in a deductive sense. This leads to the introduction of *mediated attacks* as an intermediate step for the evaluation of the framework. The intuition behind these attacks is that if from a we can deduce b, then if we do not have b, we also cannot have a. These attacks then allow for a BAF to be evaluated using standard abstract argumentation tools, for example by using the complete semantics to identify the sets of acceptable arguments.

Formally, they define the semantics of bipolar argumentation frameworks with respect to their flattening. The flattened framework will consist of meta-arguments and an attack relation only, with the deductive support relation from the BAFs being represented as a combination of auxiliary meta-arguments and attack relations.

Definition 2.1.20 (BAF Meta-arguments)**.** Given a bipolar argumentation framework $\langle \mathcal{A}, \rightarrow, \Rightarrow_d \rangle$, the set of corresponding meta-arguments MA is $\{acc(a) \mid a \in \mathcal{A}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{A}$ s.t. $a \Rightarrow_d b\}$ and $\rightarrow_2 \subseteq MA \times MA$ is a binary relation on MA such that:

- For all $a, b \in \mathcal{A}$ such that $a \Rightarrow_d b$, we have $acc(b) \rightarrow_2 Z_{a,b}$ and $Z_{a,b} \rightarrow_2 acc(a)$

Example 2.1.2. The example represented in Figure 2.2 is flattened in Figure 2.3:

Let us now examine the case of necessary support, as described in [30].

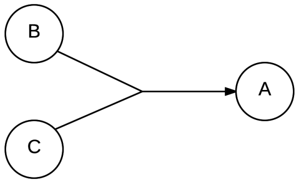


Figure 2.4: Higher level argumentation framework

Definition 2.1.21 (Argumentation Framework with Necessities (AFN))**.** An *argumentation framework with necessities* (AFN) is a triple $\langle \mathcal{A}, \rightarrow, \Rightarrow_n \rangle$ where $\mathcal{A}$ is a set of arguments, $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation and $\Rightarrow_n \subseteq \mathcal{A} \times \mathcal{A}$ is a necessity relation.

Auxiliary attacks are also defined here in terms of the support relation. In this case, they are called extended attacks. The behavior of this relation is similar to the one of the sub-argument relation in structured argumentation frameworks such as ASPIC+ [6]: if a_1 supports a_2, then attacking a_1 will result in an extended attack on a_2.

The semantics are then defined in a similar way as for deductive support, by making these extended attacks explicit and then applying standard abstract argumentation tools.

The behavior of this support relation can be replicated inside a standard AF via flattening, as described in [26]. It is quite similar to the behavior of deductive support, just in the opposite direction.

Definition 2.1.22 (AFN Meta-Arguments)**.** Given an AFN $\langle \mathcal{A}, \rightarrow, \Rightarrow_n \rangle$, the set of corresponding meta-arguments MA is $\{acc(a) \mid a \in \mathcal{A}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{A} \text{ s.t. } a \Rightarrow_n b\}$ and $\rightarrow_2 \subseteq MA \times MA$ is a binary relation on MA such that:

- For all $a, b \in \mathcal{A}$ such that $a \Rightarrow_n b$, we have $acc(a) \rightarrow_2 Z_{a,b}$ and $Z_{a,b} \rightarrow_2 acc(b)$

2.1.5 Joint attacks

Another extension of AFs allows for joint attacks, where multiple arguments join forces to attack another argument.

D. Gabbay [24] calls this kind of relation a *joint attack*. He defines it as follows:

Definition 2.1.23 (Higher Level Argumentation Framework)**.** A *higher level argumentation framework* is a triple $\langle \mathcal{A}, \mathcal{A}^0, \rightarrow \rangle$, where $\mathcal{A} \neq \emptyset$ is a set of arguments, $\mathcal{A}^0$ is the family of all finite non-empty subsets of $\mathcal{A}$ and $\rightarrow \subseteq \mathcal{A}^0 \times \mathcal{A}$ is an attack relation.

For simplicity of notation we will sometimes write x for the singleton set $\{x\}$ when this does not cause any ambiguities.

Similarly as before, the semantics of higher level frameworks will be defined in terms of their flattening. We define the flattening as follows:

Definition 2.1.24 (Higher Level AF Meta Arguments)**.** Given a higher level argumentation framework $\langle \mathcal{A}, \mathcal{A}^0, \rightarrow \rangle$, the set of corresponding meta-arguments MA is $\{acc(a), rej(a) \mid a \in \mathcal{A}\} \cup \{e(X) \mid X \in \mathcal{A}^0\}$ and $\rightarrow_2 \subseteq MA \times MA$ is a binary relation on MA such that:

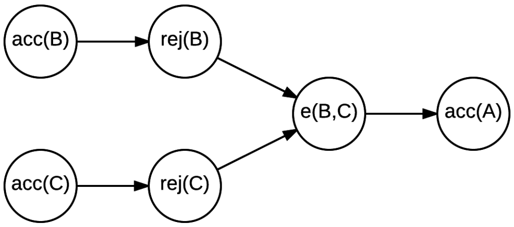


Figure 2.5: Flattened version of the framework from Figure 2.4

- For all $a \in \mathcal{A}$, we have $acc(a) \rightarrow_2 rej(a)$
- For all $X \in \mathcal{A}^0$, and every $b \in \mathcal{A}$ such that $X \rightarrow b$, we have that $e(X) \rightarrow_2 acc(b)$ and $rej(a) \rightarrow_2 e(X)$ for every $a \in X$.

In the flattening, the success of a joint attack depends solely on the acceptance of the meta-argument $e(X)$, which itself depends on the acceptance of every argument in the coalition.

The flattening of the framework from Figure 2.4 is depicted in Figure 2.5.

2.2 The ASPIC+ Framework for Structured Argumentation

When designing an argumentation framework, two of the important design decisions which have to be made are the following: how can arguments be built and how can they be attacked? S. Modgil and H. Prakken proposed a system called the ASPIC+ framework [6] which attempts to ease the designing of argumentation models by answering those questions among others. There are two main ideas on which the ASPIC+ framework is based. The first idea is that conflicts are usually resolved with explicit preferences. The second idea is that arguments are built using either strict or defeasible inference rules. While strict rules guarantee the inference of a certain conclusion from given premises, defeasible rules only present a presumption in favor of their conclusion. The goal of the ASPIC+ framework is to provide a systematic way of constructing arguments and attacks from a knowledge base and a set of inference rules. These rules and knowledge base are more intuitive to mine from a text and easier to motivate.

In order to use the ASPIC+ system, one needs to provide some information. The first element is a logical language closed under negation. Then one has to provide two sets of (possibly empty) strict and defeasible inference rules. Additionally, one must provide a partial naming function which maps some of the defeasible rules to a formula from the chosen logical language. The collection of this information is called an argumentation system.

Definition 2.2.1 (Argumentation System)**.** An *argumentation system* is a triple $AS = (\mathcal{L}, \mathcal{R}, n)$ where:

- $\mathcal{L}$ is a logical language closed under negation ($\neg$).
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules of the form $\phi_1, ..., \phi_n \rightarrow \phi$ and $\phi_1, ..., \phi_n \Rightarrow \phi$ respectively, where ϕ_i and ϕ are well-formed formulas in $\mathcal{L}$ and $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$.
- n is a partial function such that $n : \mathcal{R}_d \longrightarrow \mathcal{L}$.

We define a function $-$ such that $-\phi = \psi$ if $\phi = \neg\psi$, otherwise $-\phi = \neg\phi$.

The intuition is that the rules in $\mathcal{R}$ are on the meta-level compared to the language $\mathcal{L}$ and allow one to conclude the head of a rule if given the antecedents. The strict rules are rules of inference which are considered to hold in all cases. Hence, if one accepts its antecedents, then one must also accept its conclusion. Defeasible rules on the other hand are ones which are known to be generally true but which might fail in some cases and hence their inferences and conclusions are possible subjects of attacks.

For the rules to be of some use, one needs to define a set of premises which will serve as a knowledge base from which one can start building arguments by applying the rules.

Definition 2.2.2 (Knowledge Base)**.** A emphknowledge base in an $AS = (\mathcal{L}, \mathcal{R}, n)$ is a set of $\mathcal{K} \subseteq \mathcal{L}$ consisting of two disjoint subsets $\mathcal{K}_n$ (the axioms) and $\mathcal{K}_p$ (the ordinary premises).

The axioms are formulas of which the truth value is indisputable, while the ordinary premises are formulas which can be used to make further inferences but might turn out to be defeated in the end. By joining these with an appropriate argumentation system, one gets an argumentation theory.

Definition 2.2.3 (Argumentation Theory)**.** An emphargumentation theory is a tuple $AT = (AS, \mathcal{K})$ where AS is an argumentation system and $\mathcal{K}$ is a knowledge in AS.

An argumentation theory now contains all the elements needed for building the arguments. ASPIC+ provides a few ways to construct these from the theory. An argument in ASPIC+ has a few properties which are given by the following functions:

- Prem returns the set of all ordinary premises of the argument.
- Conc returns the conclusion of the argument.
- Sub returns all its sub-arguments.
- DefRules returns the set of defeasible rules used in the argument.
- TopRule returns the last inference rule used, if applicable.

We now have three different ways to build an argument. Either it introduces one of the ordinary premises from the knowledge, or it makes an inference from a strict or defeasible rule.

Definition 2.2.4 (ASPIC+ Argument)**.** An emphargument a on the basis of an argumentation theory with a knowledge base $\mathcal{K}$ and an argumentation system $(\mathcal{L}, \mathcal{R}, n)$ has one of the following forms:

1. φ, where $\varphi \in \mathcal{K}$ with:
   $\mathsf{Prem}(a) = \{\varphi\}$,
   $\mathsf{Conc}(a) = \varphi$,
   $\mathsf{Sub}(a) = \{\varphi\}$,,
   $\mathsf{DefRules}(a) = \emptyset$,,
   $\mathsf{TopRule}(a)$ is undefined.

2. $a_1, ..., a_n \rightarrow \varphi$, where $a_1, ..., a_n$ are arguments such that $\mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n) \rightarrow \varphi \in \mathcal{R}_s$ with:
   $\mathsf{Prem}(a) = \mathsf{Prem}(a_1) \cup ... \cup \mathsf{Prem}(a_n)$,
   $\mathsf{Conc}(a) = \varphi$,
   $\mathsf{Sub}(a) = \mathsf{Sub}(a_1) \cup ... \cup \mathsf{Sub}(a_n) \cup \{a\}$,
   $\mathsf{DefRules}(a) = \mathsf{DefRules}(a_1) \cup ... \cup \mathsf{DefRules}(a_n)$,
   $\mathsf{TopRule}(a) = \mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n) \rightarrow \varphi$.

3. $a_1, ..., a_n \Rightarrow \varphi$, where $a_1, ..., a_n$ are arguments such that $\mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n) \Rightarrow \varphi \in \mathcal{R}_d$ with:
   $\mathsf{Prem}(a) = \mathsf{Prem}(a_1) \cup ... \cup \mathsf{Prem}(a_n)$,
   $\mathsf{Conc}(a) = \varphi$,
   $\mathsf{Sub}(a) = \mathsf{Sub}(a_1) \cup ... \cup \mathsf{Sub}(a_n) \cup \{a\}$,
   $\mathsf{DefRules}(a) = \mathsf{DefRules}(a_1) \cup ... \cup \mathsf{DefRules}(a_n) \cup \{\mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n) \Rightarrow \varphi\}$,
   $\mathsf{TopRule}(a) = \mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n) \Rightarrow \varphi$.

Example 2.2.1. Consider a knowledge base in an argumentation system with language $\mathcal{L}$ consisting of p, q, r, s, t, d_1, d_2 and their negations, with $\mathcal{R}_d = \{d_1, d_2\}$ and $\mathcal{R}_s = \{s_1, s_2\}$, where the rules are defined as:

- d_1: $r \Rightarrow \neg q$
- d_2: $t \Rightarrow \neg p$
- s_1: $p \rightarrow q$
- s_2: $s \rightarrow \neg d_2$

Also, the knowledge base is formed by $\mathcal{K}_n = \{r, s\}$ and $\mathcal{K}_p = \{p, t\}$. Notice that we have defined rules by writing them in the form $n(r) : r$.

Two of the arguments we can construct are $a_1 = p$ and $a_2 = a_1 \rightarrow q$, where $\mathsf{Prem}(a_2) = \{p\}$, $\mathsf{Conc}(a_2) = q$, $\mathsf{Sub}(a_2) = \{a_1, a_2\}$, $\mathsf{DefRules}(a_2) = \emptyset$, $\mathsf{TopRule}(a_2) = s_1$.

Now that we have defined a way to construct the arguments from the knowledge base and inference rules, we can define how to build the other component of an abstract argumentation framework, namely the attacks. There are 3 ways for an argument to attack another one. It must attack it either on one of its premises, on the inference rule used or on the conclusion.

Definition 2.2.5 (ASPIC+ Attack)**.** An argument a emphattacks an argument b if and only if a *undermines, undercuts* or emphrebuts b, where:

- a emphundermines b on φ if and only if $\mathsf{Conc}(a) = -\varphi$ for an ordinary premise $\varphi \in \mathsf{Prem}(b)$.
- a emphundercuts b (on b') if and only if $\mathsf{Conc}(a) = -n(r)$ for some $b' \in \mathsf{Sub}(b)$ such that $\mathsf{TopRule}(b') = r$.
- a emphrebuts b (on b') if and only if $\mathsf{Conc}(a) = -\varphi$ for some $b' \in \mathsf{Sub}(b)$ of the form $b''_1, ..., b''_n \Rightarrow \varphi$.

Example 2.2.2. In our previous example, we can also construct the arguments $b_1 = r$, $b_2 = b_1 \Rightarrow \neg q$, $c_1 = t$, $c_2 = c_1 \Rightarrow \neg p$, $e_1 = s$ and $e_2 = e_1 \rightarrow \neg d_2$. We then have that c_2 undermines a_1 and a_2 on p, a_2 rebuts b_2 and e_2 undercuts c_2.

Notice that rebuttal is usually symmetric, however this kind of duality might be resolved by having a preference over the arguments. This way, an argument may only attack another one if it is at least as preferred as the attacked one. Given a preference relation, we can then define what it means for an attack be successful, and in general we will say that an argument a emphdefeats an argument b if the attack is successful.

Definition 2.2.6 (Successful Undermining, Rebuttal and Defeat)**.** Given a preference relation $\preceq$ over the arguments, we say that:

- a emphsuccessfully undermines b if and only if a undermines b on φ and $\varphi \preceq a$.
- a emphsuccessfully rebuts b if and only if a rebuts b on b' and $b' \preceq a$.
- a emphdefeats b if and only if it undercuts, successfully undermines or successfully rebuts b.

Notice that all undercuttings are considered as successful irrespective of preference as no such criteria is required for that kind of attack.

We can then define the procedure to generate an abstract argumentation framework from an argumentation theory and a preference relation.

Definition 2.2.7 (Corresponding AF)**.** An *abstract argumentation framework* (AF) *corresponding to* an argumentation theory $AT = (AS, \mathcal{K})$ and a preference relation over arguments $\preceq$ is a pair $(\mathcal{A}, \rightarrow)$, such that:

- $\mathcal{A}$ is the smallest set of all finite arguments constructed from $\mathcal{K}$ in AS satisfying Definition 2.2.4;
- $(x, y) \in\rightarrow$ if and only if x defeats y with respect to $\preceq$.

The preference relation is easier to motivate and understand if it is first defined on the set of defeasible rules and premises. We can then lift the preference relation from rules to arguments in one of several ways. One is the weakest-link principle, another is the last-link principle. In the weakest-link principle we compare two arguments a and b by comparing the least preferred rules in $\mathsf{DefRules}(a)$ and $\mathsf{DefRules}(b)$. If the least preferred rule in $\mathsf{DefRules}(a)$ is at least as preferred as the weakest rule in $\mathsf{DefRules}(b)$, then we say that a is at least as preferred to b. Formally, we get:

Definition 2.2.8 (Weakest-link preference)**.** Let a and b be two arguments. We have that $a \preceq_w b$ if and only if:

1. If $\mathsf{DefRules}(a) = \mathsf{DefRules}(b) = \emptyset$, then there exists $p_a \in \mathsf{Prem}(a)$, such that for all $p_b \in \mathsf{Prem}(b)$, we have $p_a \leq p_b$, else;
2. If $\mathsf{Prem}(a) = \mathsf{Prem}(b) = \emptyset$, then there exists $r_a \in \mathsf{DefRules}(a)$, such that for all $r_b \in \mathsf{DefRules}(b)$, we have $r_a \leq r_b$, else;
3. There exists $r_a \in \mathsf{DefRules}(a)$ and $p_a \in \mathsf{Prem}(a)$, such that for all $r_b \in \mathsf{DefRules}(b)$ and $p_b \in \mathsf{Prem}(b)$, we have $r_a \leq r_b$ and $p_a \leq p_b$

We define a notion of strict preference $\prec_w$ by replacing $\leq$ with $<$ in the above definition.

The other way to lift a preference relation over rules to one over arguments is by using the last link principle. According to this principle, we compare the last defeasible rules used in the argument, which corresponds to the value given by applying the function LastDefRules to the argument. We define this function as follows:

Definition 2.2.9 (Last defeasible rules)**.** Let a be an argument. We define the function LastDefRules as follows:

- If DefRules(a) = $\emptyset$, then LastDefRules(a) = $\emptyset$, else;
- If $a = a_1, ..., a_n \Rightarrow \varphi$, then $\mathsf{LastDefRules}(a)$ = $\{\mathsf{Conc}(a_1), ..., \mathsf{Conc}(a_n)\}$, else;
- If $a = a_1, ..., a_n \rightarrow \varphi$, then $\mathsf{LastDefRules}(a)$ = $\{\mathsf{LastDefRules}(a_1), ..., \mathsf{LastDefRules}(a_n)\}$

We then define the lifting of the preference from rules to arguments according to last link principle as:

Definition 2.2.10 (Last link preference)**.** Let a and b be two arguments. We have that $a \preceq_l b$ if and only if:

- If $\mathsf{LastDefRules}(a) = \mathsf{LastDefRules}(b) = \emptyset$, then there exists $p_a \in \mathsf{Prem}(a)$ such that for all $p_b \in \mathsf{Prem}(b)$, we have $p_a \leq p_b$, else;
- There exists $r_a \in \mathsf{LastDefRules}(a)$ such that for all $r_b \in \mathsf{LastDefRules}(b)$, w have $r_a \leq r_b$.

Again, we define the strict preference relation $\prec_l$ by replacing $\leq$ with $<$ in the above definition.

Chapter 3

Commitments in argumentation

3.1 Introduction

Given that many argumentation semantics have been proposed in the literature [9] and that most argumentation semantics allow for multiple extensions [10], applications of abstract argumentation theory are faced with two choice problems: First, how to choose among the various argumentation semantics? Second, given an argumentation semantics, how to choose an extension?

An important methodology to support rational choices concerning the first problem is the principle-based approach [11, 12, 10]. In this chapter, we propose a novel methodology to support choice-making concerning the second problem, i.e. concerning the selection of one among many extensions of a given AF in a given semantics.

Sometimes the need to choose an extension is circumvened by merging all extensions into a single justification status for each argument [32, 9]. For example, an argument is said to be *strongly accepted* iff it is in all extensions. However, this approach gives up the desirable properties of extensions that have been built into the chosen semantics. For example, the set of strongly accepted arguments in preferred semantics may not be admissible. This problem can be avoided by choosing one extension rather than merging all extensions into a single justification status. But this makes the question of how to choose among multiple extensions a very pressing question.

In this chapter, we do not favor one particular method for choosing an extension, but instead propose a methodology for studying and analysing this problem. We introduce the notion of a *commitment graph*, to represent the commitment towards a single extension from a set of them for a given AF in a given semantics. The edges of a commitment graph represent crucial commitment points that bring one closer to the final choice of a single extension. We distinguish between *abstract commitment graphs*, where the only content present in the nodes is extension-labels on the leaves, and instantiations of these with *concrete commitment graphs* that give a particular meaning to every node of the graph. Just like the distinction that Dung [4] introduced between abstract argumentation frameworks and structured instantiations thereof, this distinction helps to distill the features of commitment graphs that come from the graph structure alone and study these separately.

In this chapter we do not propose to extend Dung's notion of argumentation frameworks. Dung has been criticized for its abstract nature and therefore Dung's formalism has been generalized in many ways, for example with structured frameworks [6], ADFs [33], etc. Such extensions are outside the scope of this chapter, but some interesting possibilities suggested by our work are discussed in the future work section of this chapter. Instead, in this chapter we give a new perspective on existing abstract argumentation semantics in terms of commitment graphs.

Furthermore, note that our choice to base commitment graphs on the traditional extension-based approach to abstract argumentation semantics rather than on the labeling-based approach [13, 9] is merely due to the fact that this simplifies the exposition of our ideas. All the ideas developed in this chapter could also be developed with respect to the labelling-approach, and actually this would give rise to more fine-grained commitment graphs, which allow for more flexibility in the choice-making process. We leave the exploration of this adaptation of our ideas to future work.

Consider the framework depicted in Fig. 3.1 (a). With preferred semantics, we have three possible extensions: $\{a, c\}$, $\{a, d\}$ and $\{b, d\}$. In the commitment graph depicted in Fig. 3.1 (b), we see that one can either first commit to accepting a and rejecting b, or to accepting d and rejecting c, and then from there reach any of the extensions in the leaves of the graph. The argumentation framework depicted in Fig. 3.1 has the same three preferred extensions, and therefore the same commitment graph in Fig. 3.1 (b) represents the possible gradual commitments. This means that for the sake of summarization, the framework in (c) corresponds to the same commitment graph while containing one less attack, and it is therefore more compact. Notice that this framework cannot be further reduced without changing the commitment structure.

Now consider a scenario where we wish to enforce a kind of directionality in the commitment process, so that in Fig. 3.1, since elements of the pair $\{a, b\}$ have a path to elements of the pair $\{c, d\}$ but not vice-versa, we cannot commit on the status of c or d unless we have already committed towards some status on a and b. In that case, we obtain the commitment graph in Fig. 3.1 (d), where one cannot first commit on rejecting c and accepting d while keeping the option to accept or reject a. So committing towards the acceptance of d leads to accepting b as well, since the other extension where b is rejected is reachable in smaller steps by first accepting a and only then accepting d. Applying this requirement to the first framework does not change anything however, since every element has a path to every other one. In terms of summarization however, this means that the two frameworks from (a) and (c) do not have the same commitment structure anymore, and therefore the framework from (a) cannot be summarized by the one from (c). It turns out that under these conditions, the framework from (a) cannot be summarized further.

The layout of this chapter is as follows. In Section 3.2 we introduce *abstract commitment graphs* as well as the notion of *commitment mappings* that map each AF to an abstract commitment graph. Inspired by the principle-based approach to argumentation theory, we additionally define in this section two principles of commitment mappings that seem desirable, the principle of commitment-graph directionality and the one of directional choice-making. In Section 3.3 we define a first concrete instantiation of commitment graphs, namely *most fine-grained commitment graphs*, whose corresponding commitment mapping satisfies one of the two principles from Section 2 and does not satisfy the other. In Section 3.4 we introduce an alternative instantiation of commitment graphs called *SCC-*

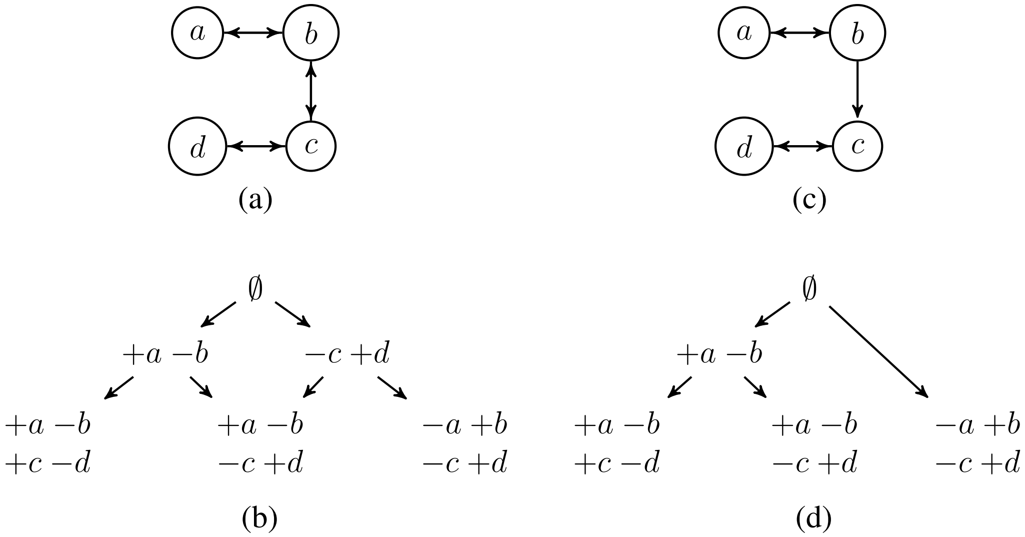


Figure 3.1: (a) Example argumentation framework F. (b) Commitment graph of F with respect to preferred semantics. (c) F', sub-framework of F. (d) Directional commitment graph of F' with respect to preferred semantics.

directional commitment graphs that is based on the well-known SCC-recursive scheme and that satisfies both principles from Section 3.2. In Section 3.5 we present a different kind of commitment graphs based on the distance between extensions. In Section 3.6, we provide equivalence and summary notions for abstract argumentation frameworks based on commitment graphs. In Section 3.7 we discuss related work, and in Section 3.8 we conclude and discuss topics for further research.

3.2 Abstract commitment graphs

In this section, we introduce *abstract commitment graphs*, where the only content present in the nodes is extension-labels on the leaves. We do however have a few requirements on the graph: It should be a directed acyclic graph, with a single root from which all other nodes are reachable, to represent our starting point in the choice-making process. Also, we require that each node connect to a distinct set of reachable endpoints, since we are interested in the processes where some extensions are discarded at every step as we traverse the graph.

We borrow the well-studied concept of rooted graph [34], also sometimes called flow-graph [35].

Definition 3.2.1 (Directed Rooted Graph)**.** A *directed rooted graph* is a directed graph (V, E) with an element $r \in V$ (called the *root*) such that there is a path from r to every other element of V. A *labeled directed rooted graph* is a triple (V, E, L) where (V, E) is a directed rooted graph and L is a (partial) labeling of V. Given $c \in V$, we define *reachable-leaves*$(c) := \{c' \in V \mid c'$ is a leaf of (V, E) and c' is E-reachable from $c\}$.

We then extend this notion to the new notions of abstract extension graph and abstract commitment graph.

Definition 3.2.2 (Abstract Extension Graph, Abstract Commitment Graph)**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$, we say that a labeled directed rooted graph (V, E, L) is an *abstract extension graph* of F iff (V, E) is acyclic and L assigns distinct subsets of $\mathcal{A}$ to the leaves of (V, E). Additionally, if for all distinct $c, c' \in V$ we have *reachable-leaves*$(c) \neq$ *reachable-leaves*(c'), then we say that (V, E, L) is an *abstract commitment graph*. In this case, we call the elements of V *commitment points*.

Example 3.2.1. Consider the argumentation framework depicted in Fig. 3.2.(a). A possible abstract commitment graph with respect to preferred semantics is the one depicted in Fig. 3.2.(b).

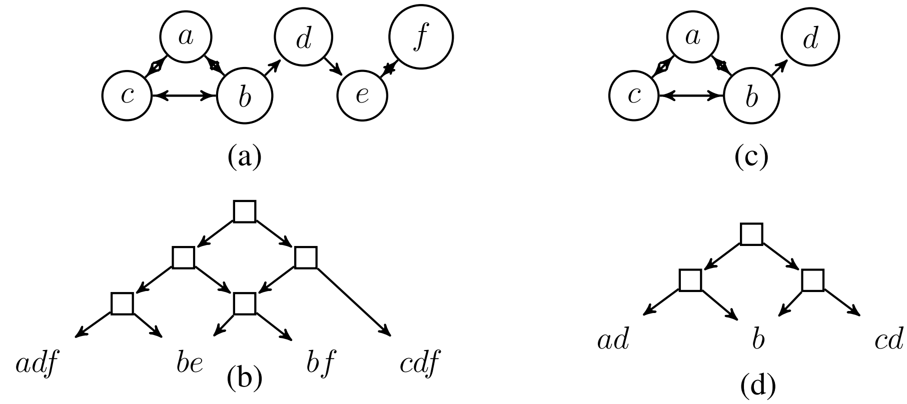


Figure 3.2: (a) Example argumentation framework $F_1 = \langle \mathcal{A}_1, \rightarrow_1 \rangle$. (b) A possible abstract commitment graph G of F_1 with respect to preferred semantics. (c) An unattacked subframework $F_1' = \langle \mathcal{A}_1', \rightarrow_1' \rangle$ of F_1, where $A_1' = \{a, b, c, d\}$. (d) The restriction $G \downarrow_{\mathcal{A}_1'}$.

We now wish to examine some properties of functions which return abstract commitment graphs for any AF, which we call *commitment mappings*.

Definition 3.2.3 (Commitment Mapping)**.** Let $\mathbb{F}$ be the class of all argumentation frameworks and $\mathbb{C}$ the class of all abstract commitment graphs. A *commitment mapping* is a function $g : \mathbb{F} \mapsto \mathbb{C}$ that for every argumentation framework F, returns an abstract commitment graph of F.

Commitment mappings can be seen as a refinement of classical abstract argumentation semantics. Instead of returning a set of extensions for a given graph, an abstract commitment graph is returned instead, providing more granularity between the framework and the different extensions. We thus introduce a notion of correspondence between commitment mappings and semantics by saying that a commitment mapping can give rise to a semantics by providing commitment graphs where the leaves are exactly the extensions that semantics would return.

Definition 3.2.4 (Giving Rise to a Semantics σ_g)**.** Given a commitment mapping g, we say that g *gives rise* to the semantics σ_g, defined such that for all argumentation frameworks $F \in \mathbb{F}$, $\sigma_g(F) = \{L(n) \mid n \text{ is a leaf in } (V, E, L), \text{ where } g(F) = (V, E, L)\}$.

One important principle studied in the principle-based approach to argumentation theory is the Principle of Directionality, which was introduced by Baroni and Giacomin [11], and which has been extensively studied for abstract argumentation semantics [10]. We now propose a way to translate this principle to a similar principle for commitment mappings.

We start with the notion of *unattacked sub-framework*, which is a sub-framework such that no argument outside of it attacks an argument inside of it. In terms of directionality, these are sub-frameworks that one should be able to evaluate locally, i.e. without having to take into account the rest of the framework. This formal notion is also sometimes called *initial* in the literature [36].

Definition 3.2.5 (Unattacked Sub-framework)**.** We say that $F' = \langle \mathcal{A}', \rightarrow' \rangle$ is an *unattacked sub-framework* of $F = \langle \mathcal{A}, \rightarrow \rangle$ iff $\mathcal{A}' \subseteq \mathcal{A}$, $\rightarrow' = \rightarrow \cap \mathcal{A}' \times \mathcal{A}'$ and there is no argument $a \in \mathcal{A} \setminus \mathcal{A}'$ attacking some argument $b \in \mathcal{A}'$.

We define a notion of equivalence between commitment points based on whether their reachable endpoints are equal with respect to the sub-framework of interest. This allows us then to define a notion of a contraction of a commitment graph, so that it only represents choices made on a sub-framework of the original one while ensuring it still satisfies the requirements for an abstract commitment graph.

Definition 3.2.6 (Commitment-equivalence)**.** Given an abstract commitment graph (V, E, L) of an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$, a subset of arguments $\mathcal{A}' \subseteq \mathcal{A}$ and two commitment points $c_1, c_2 \in V$, we say that c_1 and c_2 are commitment-equivalent with respect to $\mathcal{A}'$ (denoted as $c_1 \simeq_{\mathcal{A}'} c_2$) iff $\{L(c) \cap \mathcal{A}' \mid c \in \textit{reachable-leaves}(c_1)\} = \{L(c) \cap \mathcal{A}' \mid c \in \textit{reachable-leaves}(c_2)\}$.

Definition 3.2.7 (Restriction)**.** Given an abstract commitment graph $G = (V, E, L)$ of an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a subset of arguments $\mathcal{A}' \subseteq \mathcal{A}$, we define the *restriction* of G to $\mathcal{A}'$ as $G \downarrow_{\mathcal{A}'} = (V', E', L')$, where:

1. V' is the set of equivalent classes of $\simeq_{\mathcal{A}'}$ in G;

2. $(c, c') \in E'$ iff $c \neq c'$ and $\exists c_1 \in c, c_2 \in c'$ such that $(c_1, c_2) \in E$;

3. for every leaf c, $L'(c) = L(c_1) \cap \mathcal{A}'$ where $c_1 \in c$ is a leaf in G.

Lemma 3.2.1. *Given an abstract commitment graph G of an argumentation framework $\langle \mathcal{A}, \rightarrow \rangle$ and a set $\mathcal{A}' \subseteq \mathcal{A}$, the restriction $G \downarrow_{\mathcal{A}'}$ is also an abstract commitment graph.*

Example 3.2.2. Fig. 3.2.(c) depicts an unattacked sub-framework $F_1' = \langle \mathcal{A}', \rightarrow' \rangle$ of the framework F_1 depicted in Fig. 3.2.(a). The restriction of the commitment graph in 3.2.(b) to $\mathcal{A}'$ is depicted in Fig. 3.2.(d).

We can now define our principle of directionality for commitment mappings:

Definition 3.2.8 (Commitment-graph directionality)**.** We say that a commitment mapping g satisfies *commitment-graph directionality* iff for any argumentation frameworks $F = \langle \mathcal{A}, \rightarrow \rangle$ and $F' = \langle \mathcal{A}', \rightarrow' \rangle$ such that F' is an unattacked sub-framework of F, $g(F') = g(F) \downarrow_{F'}$.

Another principle can be derived from the notions defined above, also on the topic of directionality, but this time also incorporating the ideas of choice-making. The idea here is that the commitments should follow the directionality of the graph, so that if an argument a can reach another argument b but not vice-versa, then commitments on the status of a should come no later than commitments on the status of b. For this we first define what it means for one to be committed on the status of an argument at a given commitment point.

Definition 3.2.9 (Committed arguments ($\mathcal{A}_c$))**.** Given a commitment graph (V, E, L) of an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a commitment point $c \in V$, we define the set $\mathcal{A}_c$ of arguments whose status is committed in c to be $\{a \in \mathcal{A} \mid \text{either} \, \forall c' \in \textit{reachable-leaves}(c).\ a \in L(c'),\ \text{or} \, \forall c' \in \textit{reachable-leaves}(c).\ a \notin L(c')\}$.

Similar to the notion of unattacked sub-framework, we then define a notion of unattacked arguments within the ones on whose status we are committed.

Definition 3.2.10 (*unattacked*(c))**.** Given a commitment graph (V, E, L) of an AF $F = \langle \mathcal{A}, \rightarrow \rangle$ and a commitment point $c \in V$, we define the set *unattacked*(c) to be $\{a \in \mathcal{A}_c \mid \nexists b \in \mathcal{A} \setminus \mathcal{A}_c$ such that there is an $\rightarrow$-path from b to a but not vice-versa $\}$.

Definition 3.2.11 (Directional choice-making)**.** We say that a commitment mapping g satisfies *directional choice-making* iff for any argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$, if $g(F) = (V, E, L)$, then for all $c \in V$:

$$\{c'' \mid c'' \in V \text{ is a leaf}, \exists c' \in \textit{reachable-leaves}(c) \text{ such that } (L(c') \cap \textit{unattacked}(c)) \subseteq L(c'')\} = \textit{reachable-leaves}(c)$$

Example 3.2.3. Consider the AF F_2 depicted in Fig. 3.3 (a), with an abstract commitment graph for it in (b). Looking at the commitment point c', we have *reachable-leaves*(c') corresponding to the extensions $\{a, d\}$ and $\{b, d\}$, and therefore $\mathcal{A}_{c'} = \{c, d, e\}$ since a is in one extension but not the other, and similarly for b. However, since for each of c, d and e, b has an $\rightarrow$-path to them but not vice-versa, we have *unattacked*$(c') = \emptyset$. So when looking at the set $\{c'' \mid c'' \in C \text{ is a leaf}, \exists c' \in \textit{reachable-leaves}(c) \text{ such that } (L(c') \cap \textit{unattacked}(c)) \subseteq L(c'')\}$, we get the set of all leaves of the commitment graph, including the one labelled ac, which is not reachable from c'. The intuition is that here one is not allowed to make commitments on c nor d while a and b are still uncommitted to. On the other hand, now looking at the commitment point c, we have *reachable-leaves*(c') corresponding to the extensions $\{a, c\}$ and $\{a, d\}$, so we have $\mathcal{A}_c = \{a, b, e\}$. For e, we still have that c has an $\rightarrow$-path to e but not vice-versa, so *unattacked*$(c) = \{a, b\}$. But now, when looking at the set $\{c'' \mid c'' \in C \text{ is a leaf}, \exists c' \in \textit{reachable-leaves}(c) \text{ such that } (L(c') \cap \textit{unattacked}(c)) \subseteq L(c'')\}$, we obtain still only the leaves which are reachable from c, since neither $\{a, c\} \cap \{a, b\}$ nor $\{a, d\} \cap \{a, b\}$ produce a subset of $\{b, d\}$. Notice that even though one is committed to the rejection of e at the commitment point c, this is not a problem. This commitment does not result from an active choice, but rather is a logical consequence of the set of extensions, since none of them contain e. Hence, we do not want to forbid all commitments which do not respect the directionality of the $\rightarrow$ relation, but only the ones resulting from a choice.

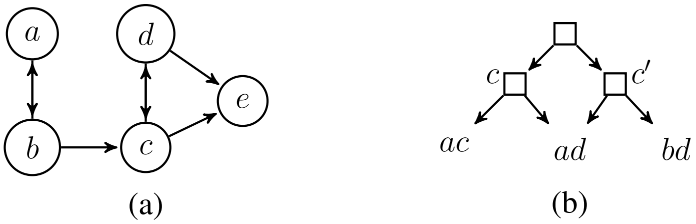


Figure 3.3: (a) Example argumentation framework F_2. (b) An abstract commitment graph for F_2.

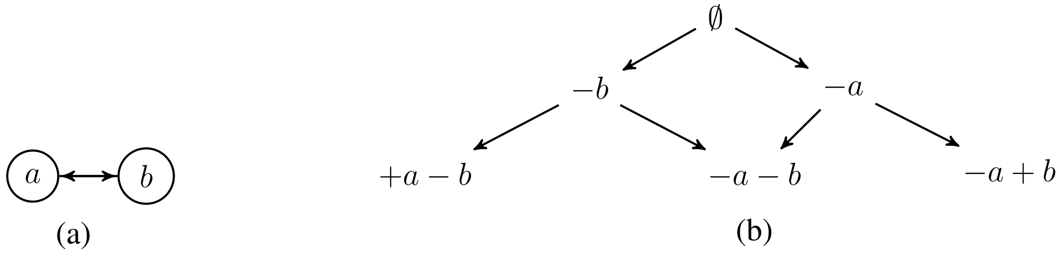


Figure 3.4: (a) Example argumentation framework F_3. (b) Intersection-based commitment graph of F_3 with respect to complete semantics, $\texttt{icg}_{\textit{complete}}(F_3)$.

3.3 Most fine-grained commitment graphs

In this section, we see an example of a commitment mapping producing *concrete commitment graphs*, where we now have labels on the intermediary points too.

We first introduce the notion of a *partial extension*, which allows to represent intermediate steps in the commitment process about which arguments to include in the extension and which arguments to exclude. Given an argument a, we denote the information that a has been chosen to be in the extension by $+a$, and the information that a has been chosen to not be in the extension by $-a$. This motivates the following definition:

Definition 3.3.1 (Partial Extension)**.** Given an argumentation framework $\langle \mathcal{A}, \rightarrow \rangle$, we define a *partial extension* for $\mathcal{A}$ to be a subset Γ of $\{+, -\} \times \mathcal{A}$ such that for no argument $a \in \mathcal{A}$, $+a \in \Gamma$ and $-a \in \Gamma$. We denote the elements of a partial extension by $+a$ and $-a$ rather than by $(+, a)$ and $(-, a)$. The set of all partial extensions for $\mathcal{A}$ is denoted by $\mathbb{P}_{\mathcal{A}}$.

When neither $+a$ nor $-a$ is in a given partial extension, this means that one has not yet committed on the status of argument a (not to be confused with the $undecided$ label from the labeling-based approach). When the status of all arguments has been determined, a *total extension* is reached:

Definition 3.3.2 (Total Extension)**.** A partial extension Γ of $\mathcal{A}$ is called a *total extension* iff for every $a \in \mathcal{A}$, either $+a \in \Gamma$ or $-a \in \Gamma$.

Example 3.3.1. Consider the framework depicted in Fig. 3.4 (a), focusing on its complete extensions $\emptyset$, $\{a\}$ and $\{b\}$.

There is a direct correspondence between the classical notion of an extension as a subset of the sets of arguments, and the notion of a total extension defined here: Having $+a$ in the total extension corresponds to a being in the corresponding extension, and having $-a$ in the total extension corresponds to a not being in the corresponding extension. This motivates the following definition:

Definition 3.3.3 (ε Function)**.** Given a set of arguments $\mathcal{A}$ and a partial extension Γ for $\mathcal{A}$, we define $\varepsilon(\Gamma) := \{a \in \mathcal{A} \mid +a \in \Gamma\}$.

ε is a bijection between the total extension for $\mathcal{A}$ and subsets of $\mathcal{A}$. So when $\mathcal{A}$ is specified, we can also refer to its inverse ε^{-1}, defined by $\varepsilon^{-1}(e) := \{+a \mid a \in e\} \cup \{-a \mid a \in \mathcal{A} \setminus e\}$.

The following notion allows us to refer to the set of arguments whose status has already been determined:

Definition 3.3.4 (Coverage)**.** Let A be a set of arguments and $\Gamma \in \mathbb{P}_{\mathcal{A}}$ be a partial extension. We define the *coverage* of Γ to be $\bar{\Gamma} := \{a \in \mathcal{A} \mid +a \in \Gamma \vee -a \in \Gamma\}$.

We now fix an argumentation framework and a semantics, allowing us to focus on the commitment structure of how the initial partial extension, from which all total extensions are reachable, leads to each one of these total extensions.

Definition 3.3.5 (Concrete Commitment Graph)**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and an abstract commitment graph (V, E, L) of F, we say that (V, E, L) is a *concrete commitment graph* (CCG) of F iff the following conditions hold:

1. $V \subseteq \mathbb{P}_{\mathcal{A}}$
2. for all leaves $c \in V$, $L(c) = \varepsilon(c)$;
3. if $(c, c') \in E$, then $c \subset c'$.

We define a straightforward concrete commitment graph where the nodes are the intersections of different subsets of extensions and the relation is just the subset relation restricted to its closest neighbor.

Definition 3.3.6 (Intersection-based commitment graph)**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a semantics σ, we define the *intersection-based commitment graph* of F with respect to σ to be $\mathtt{icg}_{\sigma}(F) = (V, E, L)$ such that $V = \{v \mid \exists X \subseteq \sigma(F), v = \bigcap\{\varepsilon(x)^{-1} \mid x \in X\}\}$, $E = \{(v, v') \mid v \subset v' \wedge \nexists v'' \in V \text{ s.t. } v \subset v'' \subset v'\}$ and L is the restriction of ε to V.

Example 3.3.2. Fig. 3.5 illustrates the notion of an intersection-based commitment graph. The preferred semantics returns three extensions, with a structure such that one can commit towards any two extensions before fully committing to a single one. Notice that if we did not differentiate partial extensions such as $-a$ from $\emptyset$, we would get a commitment graph with one less layer of granularity.

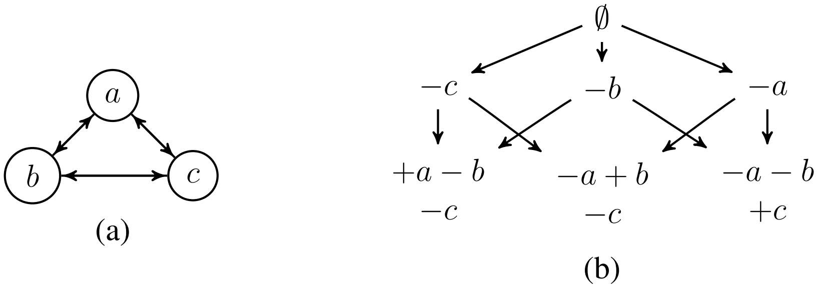


Figure 3.5: (a) Example argumentation framework F_4. (b) Intersection-based commitment graph of F_4 with respect to preferred semantics, $\mathtt{icg}_{\textit{preferred}}(F_4)$.

We now provide an equivalent, but more refined construction for the intersection-based commitment graph, which allows us to introduce additional requirements in the intermediate steps of the construction, as we describe in Section 3.4.

We first define the graph resulting from the subset relation on the partial extensions, creating an abstract extension graph. This allows us to later define a commitment mapping which constructs a concrete commitment graph with as much granularity as possible.

Definition 3.3.7 (Most Exhaustive Update (*meu*))**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a semantics σ, we define the *most exhaustive update* of F with respect to σ to be $\textit{meu}(F, \sigma) := (V, E, L)$, where $V := \{v \in \mathbb{P}_{\mathcal{A}} \mid \exists e \in \sigma(F) \text{ such that } v \subseteq \varepsilon^{-1}(e)\}$, $E := \{(v, v') \in E \mid v \subset v'\}$ and L is the restriction of ε to V.

The most exhaustive update is an abstract extension graph.

Proposition 3.3.1. *For any given argumentation framework F and semantics σ, $\textit{meu}(F, \sigma)$ is an abstract extension grah.*

We now distinguish between two kinds of edges in the most exhaustive update: the edges that relate two partial extensions that both lead to the same final total extensions, and the ones where the set of reachable total extensions becomes smaller. This corresponds to the idea that in some steps, no new information is gained, no commitments are made, and thus only reasoning is performed, while in other cases, the range of possible extensions is reduced and thus commitments are made.

Definition 3.3.8 (Reasoning Step)**.** Let $\mathcal{A}$ be a set of arguments and (V, E, L) be an abstract extension graph such that $V \subseteq \mathbb{P}_{\mathcal{A}}$. We say that $(v, v') \in E$ is a *reasoning step* iff $\textit{reachable-leaves}(v) = \textit{reachable-leaves}(v')$. Otherwise, we say that (v, v') is a *commitment step*. We denote the set of all reasoning steps in (V, E, L) by $\mathbb{R}((V, E, L))$.

We define the most fine-grained commitment graphs by focusing on the commitment steps in the most exhaustive update. For this, we need to condense the most exhaustive update such that reasoning is made automatically. This is akin to approaches in epistemic logic in which knowledge is assumed to be logically closed, i.e. in which reasoning is assumed to be instantaneously completed. We also identify the commitment points in the commitment graphs, which are the nodes where no more reasoning can be made and making a commitment cannot be avoided.

Definition 3.3.9 (Commitment Contraction)**.** Let $\mathcal{A}$ be a set of arguments, let (V, E, L) be an abstract extension graph such that $V \subseteq \mathbb{P}_{\mathcal{A}}$ and let $v \in V$. We say that $v' \in V$ is a *reasoning completion of* v *in* (V, E, L) iff v' is a $\subseteq$-maximal partial extension such that either there is an $\mathbb{R}((V, E, L))$-path from v to v', or $v = v'$. If $v \in V$ is its own reasoning completion, we say that v is a *commitment point in* (V, E, L). We denote the set of all commitment points in (V, E, L) by $\mathcal{C}((V, E, L))$. We call the graph (V', E', L'), where $V' := \mathcal{C}((V, E, L))$, $E' := E \cap (V' \times V')$ and L' is the restriction of ε to V', the *commitment contraction* of (V, E, L) and denote it by $cc((V, E, L))$.

Proposition 3.3.2. *Given an argumentation framework* $F = \langle \mathcal{A}, \rightarrow \rangle$*, a semantics* σ *and* $(V, E, L) = meu(F, \sigma)$*, for any* $v \in V$*, the reasoning completion of* v *in* (V, E, L) *is unique.*

Proof. We proceed by contradiction. Suppose there are distinct $v_1, v_2 \in S$ such that both are reasoning completions of v in (V, E, L). So both are $\subseteq$-maximal in V such that *reachable-leaves*$(v) =$ *reachable-leaves*$(v_1) =$ *reachable-leaves*(v_2). Consider the partial extension $v_3 = v_1 \cup v_2$. Clearly $v_1 \subset v_3$. Since *reachable-leaves*$(v_1) =$ *reachable-leaves*(v_2), and E is the subset relation, for every $l \in$ *reachable-leaves*(v_1), $v_3 \subseteq l$. Thus $v_3 \in V$ and *reachable-leaves*$(v_3) =$ *reachable-leaves*(v_1). So v_1 is not $\subseteq$-maximal in V such that *reachable-leaves*$(v) =$ *reachable-leaves*(v_1), and therefore we have a contradiction. □

We want to define a maximally fine-grained commitment graph, so our intention is that no commitments are skipped, however small they may be. Thus we filter out the edges which relate two nodes already connected with more fine-grained paths.

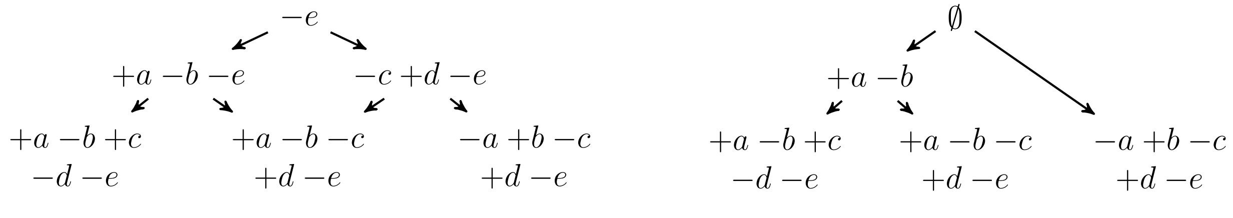
Definition 3.3.10 (Fine-grained Filtering)**.** Let $G = (V, E, L)$ be an abstract extension graph. We define the *fine-grained filtering* of G as $fgf(G) := (V, E', L)$, where $E' = \{(v, v') \in E \mid \text{there is no } E\text{-path from } v \text{ to } v' \text{ of length } > 1\}$.

Now we are ready to define the most fine-grained commitment graph of an AF with respect to a semantics:

Definition 3.3.11 (Most Fine-Grained Commitment Graph (*mfg*))**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a semantics σ, we define the *most fine-grained commitment graph* of F with respect to σ to be $mfg_\sigma(F) := fgf((V, E, L))$, where $V := \mathcal{C}(meu(F, \sigma))$, $E := \{(v, v') \in V \times V \mid v \subseteq v'\}$ and L is the restriction of ε to V.

Example 3.3.3. In Fig. 3.6, observe that the root is $-e$ instead of $\emptyset$, since $-e$ is an element of all total extensions. Also, one can see that the commitment mapping $mfg_{preferred}$ does not satisfy the principle of directional choice-making, since it is also possible to initially make a commitment on the status of the arguments c, d, e, even though there is a $\rightarrow$-path from b to all of these arguments, but not vice-versa.

We now establish the equivalence result between the simple and the more refined constructions.



(a) (b)

Figure 3.6: (a) Most fine-grained commitment graph of F_2 (see Fig. 3.3) with respect to preferred semantics, $\mathit{mfg}_{\mathit{preferred}}(F_2)$. (b) SCC-directional commitment graph of F_2 with respect to preferred semantics, $\mathit{sdcg}_{\mathit{preferred}}(F_2)$.

Lemma 3.3.3. *For any argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and semantics σ, $\mathtt{icg}_\sigma(F) = \mathit{mfg}_\sigma(F)$.*

Proof. Let $\mathtt{icg}_\sigma(F) = (V, E, L)$. We first prove that V is the set of nodes in $(V_f, E_f) = \mathit{mfg}_\sigma(F)$, i.e. V are the commitment points in $\mathit{meu}(F, \sigma)$. Let $\mathit{meu}(F, \sigma) = (V_e, E_e)$. Consider a particular $v \in V$. Then, there exists $X \subseteq \sigma(F)$ such that $v = \bigcap\{\varepsilon^{-1}(x) \mid x \in X\}$. Now, consider any strict superset v' of v. By construction of (V_e, E_e), $\mathit{reachable\text{-}leaves}(v) = \{\varepsilon^{-1}(x) \mid x \in X\}$, and E_e is the strict subset relation on V_e. Therefore, it must be that $\mathit{reachable\text{-}leaves}(v') \subset \mathit{reachable\text{-}leaves}(v)$, and hence v is its own reasoning completion. So $V \subseteq V_f$. We now show the other direction. Consider a particular $v \in V_f$, i.e. v is a commitment point in (V_e, E_e). If v is a leaf in (V_e, E_e), then we are done since it is the intersection of a singleton of extensions. If v is not a leaf, then it must be that there is no $v' \supset v$ such that $\mathit{reachable\text{-}leaves}(v') = \mathit{reachable\text{-}leaves}(v)$. Since E_e is the $\subset$ relation, it must be that $v = \bigcap \mathit{reachable\text{-}leaves}(v)$, and so there exists $X \subseteq \sigma(F)$ s.t. $v = \{\varepsilon^{-1}(x) \mid x \in X\}$. So $V = V_f$.

We now show that $E = E_f$. E_e is the subset relation on V, and E_f is the fine-grained filtering of E_e, so that $(v, v') \in E_f$ iff there is no E_e path of length ¿ 1 from v to v', i.e. there is no v'' such that $(v, v''), (v'', v') \in E_e$, or in other words $v \subset v''$ and $v'' \subset v'$. Hence $E_f = E$. □

We now state some properties of most fine-grained commitment graphs:

Theorem 3.3.4. *For any AF F and semantics σ, $\mathit{mfg}_\sigma(F)$ is a concrete commitment graph.*

Proof. Let $(V, E, L) = \mathit{mfg}_\sigma(F)$. First of all, (V, E, L) is acyclic and L assigns distinct subsets of $\mathcal{A}$ to its leaves, per Lemma 3.3.3. Additionally, per Lemma 3.3.3, one can see that the set of reachable leaves of each commitment point in V are distinct. Therefore, $\mathit{mfg}_\sigma(F)$ is an abstract commitment graph.

From Lemma 3.3.3 and the construction of $\mathtt{icg}_\sigma(F)$, one can easily see that $\mathit{mfg}_\sigma(F)$ is also a concrete commitment graph. □

Corollary 3.3.5. *For any semantics σ, mfg_σ is a commitment mapping.*

Theorem 3.3.6. *For any semantics σ that satisfies the principle of directionality defined in [11], mfg_σ satisfies the principle of commitment-graph directionality.*

Proof. This follows from the fact that σ satisfies the principle of directionality. □

Theorem 3.3.7. *For any semantics $\sigma \in \{complete, preferred, semi\text{-}stable, naive, stage, CF2, stage2\}$, mfg_σ does not satisfy the principle of directional choice-making.*

Proof. Fig. 3.6 (a) shows a counter-example for the *preferred* semantics. The same argumentation framework is a counter-example for the *complete*, *semi-stable*, *naive*, *stage*, *CF2* and *stage2* semantics. □

3.4 SCC-directional commitment graphs

In the previous section, we have seen that the most fine-grained commitment graphs do not satisfy the principle of directional choice-making. An important notion in connection with the directionality of the attack relation is the SCC-recursive schema [18], which has been used in algorithms [37, 38, 39] and in the definition of new semantics (such as the CF2 [40] and stage 2 semantics [41]).

In this section, we focus on how the SCC-recursive schema can be used as an additional layer to restrict the relation in the commitment graphs. We define the SCC-directional commitment graphs, and then prove some properties about the relation between the canonical semantics and the recursive semantics.

Recall the definition of a strongly connected component from Def. 2.1.8.

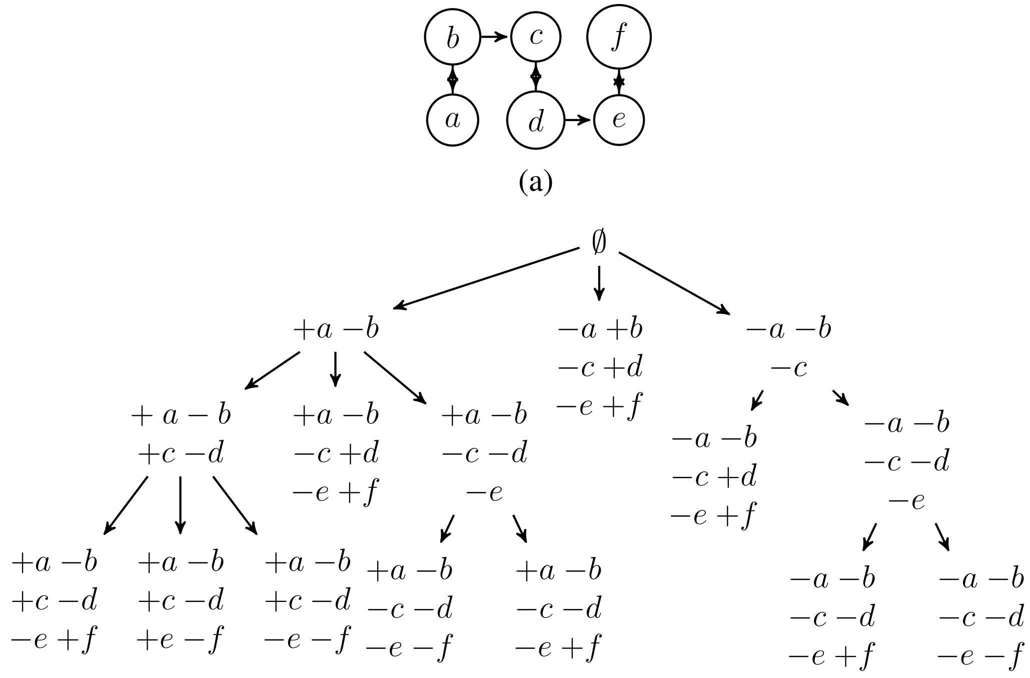
We impose a restriction on commitment graphs that only allows partial extensions to specify the status of an argument if it also specifies the status of all its SCC-ancestors (given by the $sccanc_F$ function).

Definition 3.4.1 (SCC-directionality)**.** Let F be an argumentation framework and Γ a partial extension of F. We say that Γ satisfies *SCC-directionality in F* iff for all $S \in SCCs_F$, if $S \cap \bar{\Gamma} \neq \emptyset$, then for all $S' \in sccan_F(S)$, $S' \subseteq \bar{\Gamma}$.

Definition 3.4.2 (SCC-directional update)**.** Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$ and a semantics σ such that $meu(F, \sigma) = (V, E)$, we define the *SCC-directional update* of F with respect to σ to be $sdu(F, \sigma) := (V', E')$, where $V' := \{\Gamma \in \mathbb{P}_\mathcal{A} \mid \Gamma \in V$ and Γ satisfies SCC-directionality in $F\}$ and $E' := E \cap (V' \times V')$.

Definition 3.4.3 (SCC-directional commitment graph)**.** Given an argumentation framework F and a semantics σ, we define the *SCC-directional commitment graph* of F with respect to σ to be $sdcg_\sigma(F) := fgf(cc(sdu(F, \sigma)))$.

Example 3.4.1. Consider the argumentation framework F_2 from Fig. 3.6. We have its SCC-directional commitment graph with respect to preferred semantics in Fig. 3.6 (c). Notice that the right-hand path containing $-a$ $+d$ now directly leads to a total extension, and the total extension containing $+a$ $+d$ is now only reachable by first choosing $+a$ $-b$. Also notice how in this graph the root is $\emptyset$ instead of $-e$, since even though $-e$ already follows as a reasoning step, due to e being part of a later SCC, its status is left unspecified until the one of the other arguments is determined.



(a)

(b)

Figure 3.7: (a) Example argumentation framework F_5. (b) SCC-directional commitment graph of F_5 with respect to complete semantics, $sdcg_{complete}(F_5)$.

Example 3.4.2. Consider the argumentation framework F_5 with its corresponding SCC-directional commitment graph in Fig. 3.7. Notice how the commitment towards $-a$ $+b$ in the initial SCC $\{a, b\}$ directly leads to a total extension, since $-c$ $+d$ $-e$ $+f$ immediately follow as reasoning steps, while some other paths might have more granularity since they lead to a greater number of total extensions.

We now state some properties of SCC-directional commitment graphs:

Theorem 3.4.1. *For any AF F and semantics σ, $sdcg_\sigma(F)$ is a concrete commitment graph.*

Proof. This follows in a similar way as Theorem 3.3.4. □

Corollary 3.4.2. *For any semantics σ, $sdcg_\sigma$ is a commitment mapping.*

Theorem 3.4.3. *For any semantics σ that satisfies the principle of directionality defined in [11], $sdcg_\sigma$ satisfies the principle of commitment-graph directionality.*

Theorem 3.4.4. *For any semantics σ, $sdcg_\sigma$ satisfies the principle of directional choice-making.*

3.5 Distance-based semantics

In this section, we present another instantiation of the abstract commitment graphs based on the notion of distance between extensions.

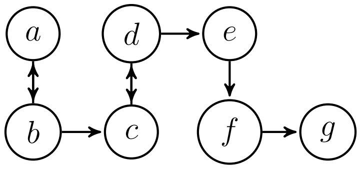


Figure 3.8: Example AF F_6.

	$\emptyset$	$\{a\}$	$\{a,c,e,g\}$	$\{a,d,f\}$	$\{b,d,f\}$	$\{d,f\}$
$\emptyset$	0	1	4	3	3	2
$\{a\}$	1	0	3	2	4	3
$\{a,c,e,g\}$	4	3	0	5	7	6
$\{a,d,f\}$	3	2	5	0	2	1
$\{b,d,f\}$	3	4	7	2	0	1
$\{d,f\}$	2	3	6	1	1	0

Table 3.1: Hamming distance table between the complete extensions of F_6, depicted in Fig. 3.8.

Definition 3.5.1 (Hamming Distance)**.** Given two sets E and E', the *hamming distance* between E and E' is $\mathit{hd}(E,E') = |(E \cup E') \setminus (E \cap E')|$.

Given a set S, a *partition* P of S is a set of subsets of S, such that $\forall s \in S, \exists p \in P$ s.t. $s \in p$ and $\forall p, p' \in P, p \cap p' = \emptyset$. We call the elements of a partition the *cells*.

We borrow the maximin approach [42] from decision theory and partition the set of extensions into at least two cells of close extensions such that the two closest elements from different cells are as distant as possible.

Definition 3.5.2 (Maximin Distance Partition)**.** Given a set of extensions S with at least two elements, we define the *maximin distance partition* of S ($mdp(S)$) to be the partition P of S such that P has as many cells as possible, at least two of them, and where $\min_{E\in p,E'\in p',p\neq p'} \mathit{hd}(E,E')$ is maximal, i.e. the distance between two closest elements from different cells is as large as possible.

Example 3.5.1. Consider the framework depicted in Fig. 3.8. The extensions are listed in Table 3.1, with their relative hamming distance to each other. Now consider the partition $\{C_1, C_2, C_3\}$ where $C_1 = \{\emptyset, \{a\}\}$, $C_2 = \{a,c,e,g\}$ and $C_3 = \{\{a,d,f\},\{b,d,f\},\{d,f\}\}$. One pair of two closest elements from different cells in this case is $\emptyset$ and $\{d,f\}$, with a hamming distance of 2. Another such pair is $\{a\}$ and $\{a,d,f\}$ with a distance of 2 as well.

We define a new kind of commitment graph where one commits to a subset of these cells in an iterative process. The intuition is that one makes big, impactful choices first, and then works out the details later.

Definition 3.5.3. Given an argumentation framework $F = \langle \mathcal{A}, \rightarrow \rangle$, a semantics σ and an abstract commitment graph (V,E,L) of F, we say that (V,E,L) is a *distance-based commitment graph* (DBCG) of F iff we have $(c,c') \in E$ iff $L(c') \in mdp(\bigcup \mathit{reachable\text{-}leaves}(c))$.

In our previous example, we obtain the commitment graph depicted in Fig. 3.9.

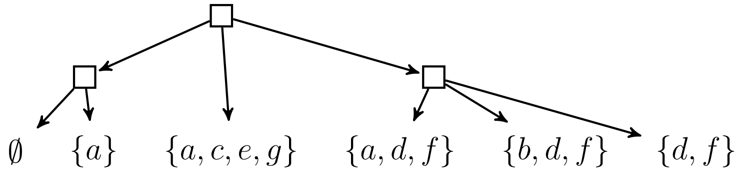


Figure 3.9: Distance-based commitment graph for the framework F_6, represented in Fig. 3.8.

3.6 Commitment graph summaries

When humans have to consider a large amount of information, it can be very helpful for them to focus on a summary of that information that contains the bits of information that are most relevant for the choices that they need to make. In this regard, it makes sense to study how AFs can be summarized in such a way that the structure of the associated commitment graph does not change.

An additional motivation for such summarization comes from structured argumentation: The AF produced from a theory of structured argumentation is often of infinite size, but when the defeasible information encoded in the theory is finite, it is usually the case that one can already commit to a specific extension of this infinite AF by committing to the status of finitely many arguments. In other words, the infinite AF resulting from the argumentation theory can be summarized to a finite AF in such a way that the structure of the associated commitment graph does not change. Let us illustrate this point on an example based on the structured argumentation framework ASPIC+ by Modgil and Prakken [43].

Example 3.6.1. Consider the language L that is the closure of $\{a, b\}$ under the negation operator $\neg$, i.e. L is the set $\{a, b, \neg a, \neg b, \neg\neg a, \neg\neg b, \dots\}$ of propositional formulas. Consider the argumentation theory that has a and $\neg b$ as ordinary premises, $a \Rightarrow b$ as the only defeasible rule and the set $\{\varphi \rightarrow \neg\neg\varphi \mid \phi \in L\}$ of strict rules.

Then the AF AF_1 associated to this argumentation theory has infinitely many arguments as follows:

$$\begin{array}{lll}
A_0 : \ a & B_0 : \ A_0 \Rightarrow b & C_0 : \ \neg b \\
A_1 : \ A_0 \rightarrow \neg\neg a & B_1 : \ B_0 \rightarrow \neg\neg b & C_1 : \ C_0 \rightarrow \neg\neg\neg b \\
A_2 : \ A_1 \rightarrow \neg\neg\neg\neg a & B_2 : \ B_1 \rightarrow \neg\neg\neg\neg b & C_2 : \ C_1 \rightarrow \neg\neg\neg\neg\neg b \\
\qquad \vdots & \qquad \vdots & \qquad \vdots
\end{array}$$

The attack relation of AF_1 is $\{(B_0, C_i) \mid i \in \mathbb{N}\} \cup \{(B_1, C_i) \mid i \in \mathbb{N}\} \cup \{(C_0, B_i) \mid i \in \mathbb{N}\}$. In the complete semantics, this AF has three extensions:

- $E_1 = \{A_i \mid i \in \mathbb{N}\}$
- $E_2 = \{A_i \mid i \in \mathbb{N}\} \cup \{B_i \mid i \in \mathbb{N}\}$
- $E_2 = \{A_i \mid i \in \mathbb{N}\} \cup \{C_i \mid i \in \mathbb{N}\}$

The choice between these three extensions depend entirely on the choice about which of B_0 and C_0 are accepted in the extension: The A_i's always get accepted. If B_0 is accepted, then all C_i's must be rejected and all other B_i's must be accepted together with B_0. Similarly, if C_0 is accepted, then all B_i's must be rejected and all other C_i's must be accepted together with C_0. And if both B_0 and C_0 get rejected, B_1 must be rejected too, for otherwise B_0 would be defended, so it would be in the complete extension; but then no other argument among the B_i's and the C_i's can be defended, so they must all be rejected.

Furthermore, note that B_0 and C_0 attack each other, so the restriction of AF_1 to $\{B_0, C_0\}$ is $AF_2 = \langle\{B_0, C_0\}, \{(B_0, C_0), (C_0, B_0)\}\rangle$, whose complete extensions are $\emptyset$, $\{B_0\}$ and $\{C_0\}$, which according to the explanations given in the previous paragraph give rise to the three extensions of AF_1.

So from the point of view of the choices between the complete extensions, the finite AF AF_2 can be considered to be a summarization of AF_1, because AF_2 is a subframework of AF_1 with the property that the choice between the extension of AF_2 correspond to choices between the extensions of AF_1.

For the purpose of defining such summarizations of AFs, we define two notions of equivalence between AFs, based on abstract and concrete commitment graphs respectively. We use these notions of equivalence to define two notions of summarization for AFs.

Definition 3.6.1 (acg-equivalence)**.** Let $F = \langle\mathcal{A}, \rightarrow\rangle$ and $F' = \langle\mathcal{A}', \rightarrow'\rangle$ be two AFs, g be a commitment mapping, $(C, R, L) = g(F)$ and $(C', R', L') = g(F')$. We say that F and F' are *acg-equivalent w.r.t.* g (written $F \simeq_g^a F'$) iff there is an isomorphism f from (C, R) to (C', R') such that for all leaves l in (C, R, L), $L(l) \cap \mathcal{A}' = L'(f(l)) \cap \mathcal{A}$.

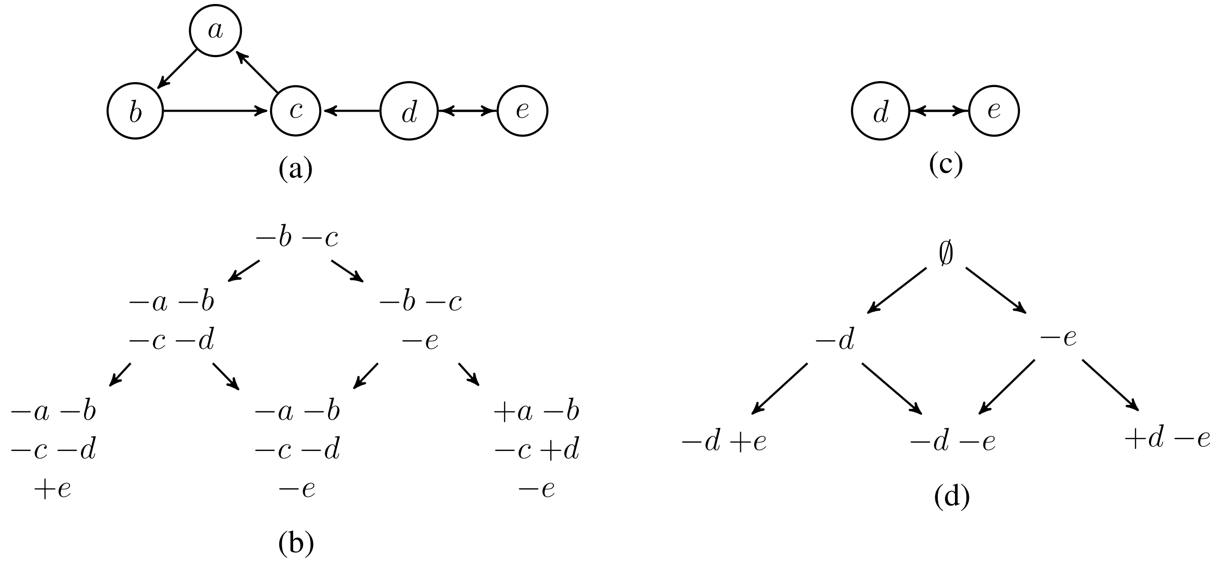
It has often been noted in the abstract argumentation literature that an odd attack path from an argument a to an argument b can be treated as an indirect attack from a to b. When summarizing argumentation frameworks, we allow such indirect attacks to be replaced by direct attacks. To formalize this, we need the following auxiliary notions:

Definition 3.6.2 (Path Length)**.** Let $F = \langle\mathcal{A}, \rightarrow\rangle$ be an AF and let $a, b \in \mathcal{A}$. An $\rightarrow$*-path from* a *to* b is a sequence $a_0, \dots, a_n$ of arguments in $\mathcal{A}$ where $a_0 = a$, $a_n = b$, a_i attacks a_{i+1} for every $0 \leq i < n$, and where $a_j \neq a_k$ for $0 \leq j < k \leq n$ with either $j \neq 0$ or $k \neq n$. The number n is called the *length* of this path.

Definition 3.6.3 (Indirect Attacks)**.** Let $F = \langle\mathcal{A}, \rightarrow\rangle$ be an AF and let $\mathcal{A}' \subseteq \mathcal{A}$. We define the set *ind*$(\rightarrow, \mathcal{A}')$ of *indirect attacks w.r.t.* $\mathcal{A}'$ to be $\{(a, b) \mid$ there is an $\rightarrow$-path from a to b of odd length such that the only arguments in this path included in $\mathcal{A}'$ are a and $b\}$.

The following definition defines the notion of an acg-summary, which formalizes summarization based on abstract commitment graphs.

Definition 3.6.4 (acg-reduction and acg-summary)**.** Let $F = \langle\mathcal{A}, \rightarrow\rangle$ and $F' = \langle\mathcal{A}', \rightarrow'\rangle$ be two AFs and g a commitment mapping. We say that F' is an *acg-reduction* of F w.r.t. g iff $\mathcal{A}' \subseteq \mathcal{A}$, $\rightarrow' \subseteq$ *ind*$(\rightarrow, \mathcal{A}')$ and $F' \simeq_g^a F$. We say that F' is *acg-irreducible* w.r.t. g iff F' is the only acg-reduction of F' w.r.t. g. We say that F' is an *acg-summary* of F wr.t. g iff F' is an acg-irreducible acg-reduction of F.



(a)

(b)

(c)

(d)

Figure 3.10: (a) Example argumentation framework F_7. (b) Most fine-grained commitment graph of F_7 with respect to complete semantics, $\mathit{mfg}_c omplete(F_7)$. (c) F_7', sub-framework of F_7. (d) Most fine-grained commitment graph of F_7' with respect to complete semantics, $\mathit{mfg}_{\mathit{complete}}(F_7')$.

Example 3.6.2. Consider F_7 depicted in Fig. 3.10(a). F_7' depicted in Fig. 3.10(c) is a acg-summary of F_7 w.r.t. $\mathit{mfg}_{\mathit{complete}}$.

Definition 3.6.5 (Coverage Restriction)**.** Let P be a partial extension and $\mathcal{A}$ a set of arguments. We define the *coverage restriction* of P to $\mathcal{A}$ as $P\downarrow_{\mathcal{A}} = \{+a \mid +a \in P, a \in \mathcal{A}\} \cup \{-a \mid -a \in P, a \in \mathcal{A}\}$.

Definition 3.6.6 (ccg-equivalence and ccg-summary)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ and $F' = \langle \mathcal{A}', \rightarrow' \rangle$ be two AFs, g be a concrete commitment mapping, $(C, R, L) = g(F)$ and $(C', R', L') = g(F')$. We say that F and F' are *ccg-equivalent w.r.t.* g (written $F \simeq_g^c F'$) iff there is an isomorphism f from (C, R) to (C', R') such that for all $d \in C$, $d\downarrow_{\mathcal{A}'} = f(d)\downarrow_{\mathcal{A}}$. We say that F' is *ccg-irreducible* w.r.t. g iff F' is the only ccg-reduction of F' w.r.t. g. We say that F' is a *ccg-summary* of F wr.t. g iff F' is a ccg-irreducible ccg-reduction of F.

Example 3.6.3. Consider the two AFs from Fig. 3.11. While $F_8 \simeq^a_{\mathit{sdcg}_{\mathit{complete}}} F_8'$, as the structure of their SCC-directional commitment graphs are the same, it is not the case that $F_8 \simeq^c_{\mathit{sdcg}_{\mathit{complete}}} F_8'$, because the content of the middle nodes does not match. While in (d) one first commits to either $-c$ or $-d$, in (b) c and d are not covered until only the last step, due to the SCC structure of F_8. However, we do have that $F_8 \simeq^c_{mfg_{\mathit{complete}}} F_8'$, since when the SCC structure is not taken into account then $-c$ and $-d$ already appear together with $-b$ and $-a$ respectively, resulting in a match between the commitment graphs of F_8 and F_8' even at the concrete level.

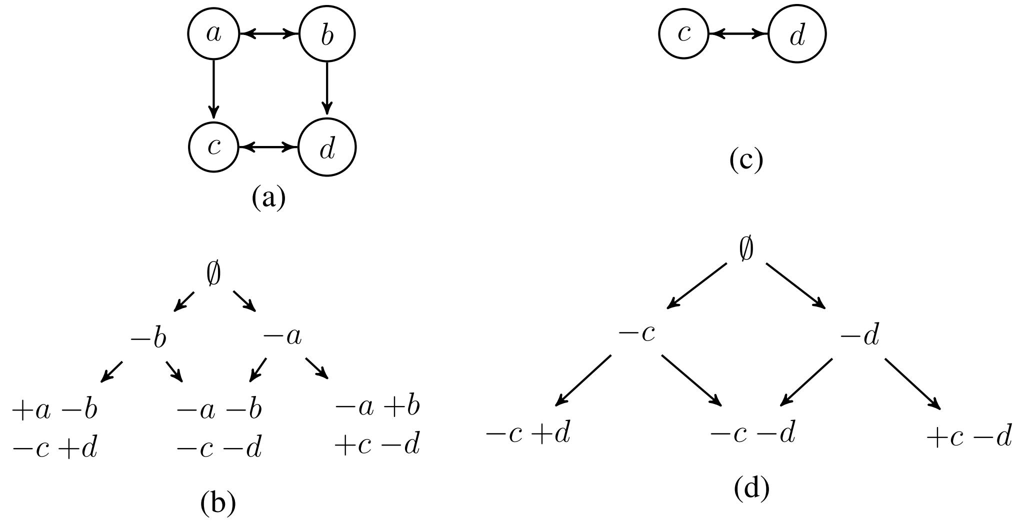


Figure 3.11: (a) Example argumentation framework F_8. (b) SCC-directional commitment graph of F_8 with respect to complete semantics, $sdcg_{complete}(F_8)$. (c) F_8', sub-framework of F_8. (d) SCC-directional commitment graph of F_8' with respect to complete semantics, $sdcg_{complete}(F_8')$.

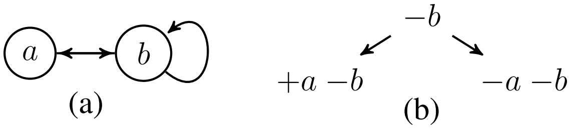


Figure 3.12: (a) Example argumentation framework F_8. (b) Most fine-grained commitment graph of F_8 with respect to complete semantics, $mfg_{complete}(F_8)$. Notice that since there is only one preferred extension for F_8, $mfg_{preferred}(F_8)$ consists of a single node. Therefore, both the acg and ccg summaries for F_8 with respect to preferred semantics are the empty framework, while the acg and ccg summaries for F_8 with respect to complete semantics are F_8 itself.

Proposition 3.6.1 (Conjecture)**.** *Given two AFs F and F', if F' is a ccg-reduction of F w.r.t. $mfg_{complete}$ then F' is a ccg-reduction of F w.r.t. $mfg_{preferred}$.*

Corollary 3.6.2. *Given an AF F, for every ccg-summary $F' = \langle \mathcal{A}', \rightarrow' \rangle$ w.r.t. $mfg_{preferred}$, there exists a ccg-summary $F'' = \langle \mathcal{A}'', \rightarrow'' \rangle$ w.r.t. $mfg_{c}omplete$ such that $\mathcal{A}' \subseteq \mathcal{A}''$ and $\rightarrow' \subseteq ind(\rightarrow'', \mathcal{A}')$.*

We present an equivalence result for the case of the most fine-grained commitment mapping.

Proposition 3.6.3. *Given a semantics σ, two frameworks F and F', F' is acg-equivalent to F with respect to mfg_σ iff F' is ccg-equivalent to F with respect to mfg_σ.*

Proof. Sketch: Since mfg_σ can be recovered from the extensions alone (see Lemma 3.3.3), i.e. the content of the leaves, the two frameworks have the same extensions (acg-equivalence) iff they have the same concrete commitment graph (ccg-equivalence). □

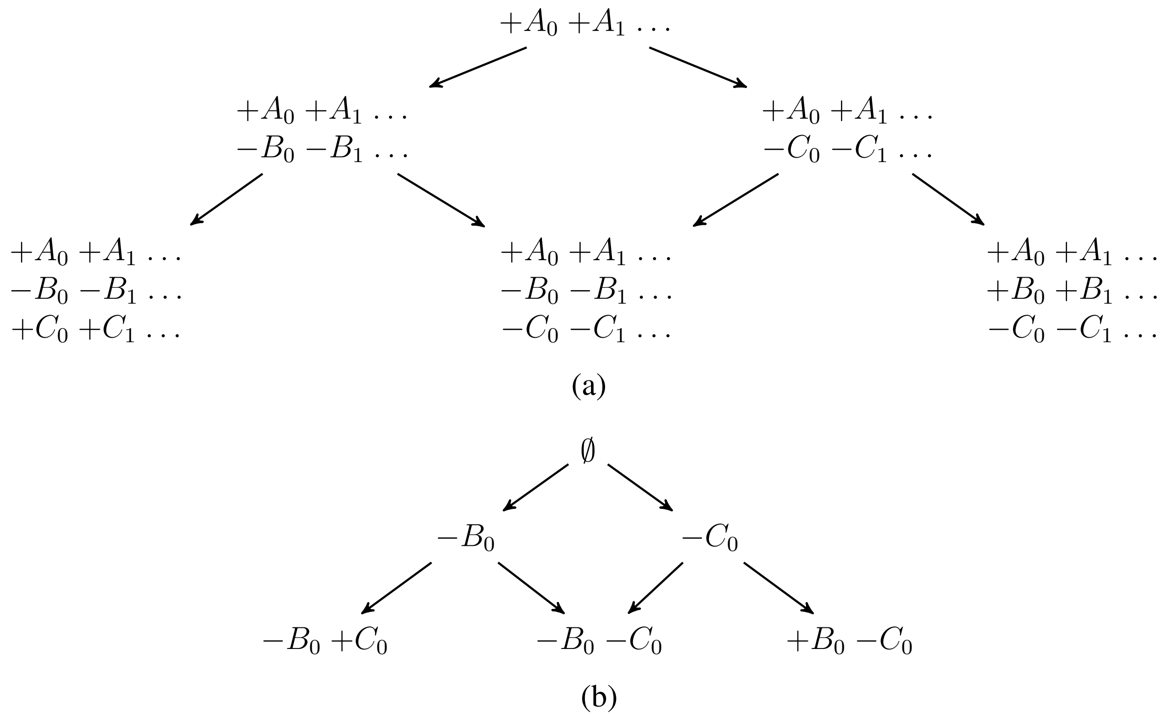

$+A_0\ +A_1 \dots$
$+A_0\ +A_1 \dots$ $-B_0\ -B_1 \dots$
$+A_0\ +A_1 \dots$ $-C_0\ -C_1 \dots$
$+A_0\ +A_1 \dots$ $-B_0\ -B_1 \dots$ $+C_0\ +C_1 \dots$
$+A_0\ +A_1 \dots$ $-B_0\ -B_1 \dots$ $-C_0\ -C_1 \dots$
$+A_0\ +A_1 \dots$ $+B_0\ +B_1 \dots$ $-C_0\ -C_1 \dots$
(a)
$\emptyset$
$-B_0$
$-C_0$
$-B_0\ +C_0$
$-B_0\ -C_0$
$+B_0\ -C_0$
(b)

Figure 3.13: (a) Most fine-grained commitment graph of F_9 with respect to complete semantics, *mfg*$_c omplete(F_9)$. (b) Most fine-grained commitment graph of F_9 with respect to complete semantics, *mfg*$_{complete}(F_9)$.

Corollary 3.6.4. *Given a semantics σ, two frameworks F and F', F' is an acg-summary of F with respect to mfg$_\sigma$ iff F' is a ccg-summary of F with respect to mfg$_\sigma$.*

We finish this section by looking back at Example 3.6.1, which we used to motivate the development of a notion of AF summary based on commitment graphs, in order to see how the introduced concepts apply to that example.

Example 3.6.4. (Continuation of Example 3.6.1.)

As can be seen in Figure 3.13, there is an isomorphism between *mfg*$_{complete}(AF_1)$ and *mfg*$_{complete}(AF_2)$ that preserves the structure of all the node labels. Thus AF_1 and AF_2 are acg-equivalent and ccg-equivalent w.r.t. *mfg*$_{complete}$. Clearly AF_2 cannot be reduced further while keeping this equivalence intact, so AF_2 is an acg-summarization and a ccg-summarization of AF_1 w.r.t. *mfg*$_{complete}$.

Similarly, one can easily see that AF_1 is an acg-summarization and a ccg-summarization of AF_1 w.r.t. *sdcg*$_{complete}$, w.r.t. *mfg*$_{preferred}$, and w.r.t. *sdcg*$_{preferred}$.

3.7 Related research

There is substantial work of applying formal argumentation theory to support decision-making [44, 45, 46]. In these papers, argumentation is used to support making decisions about other things than argumentation. That is quite different from our approach, in which we study how to theoretically study the problem of which extension to choose among multiple extensions of an AF. It remains an open problem whether our commitment graphs can be extended such that they can be applied to support decision-making as well.

Moreover, there is work on decision procedures. For example, Dvorak et al. [47] study the complexity of evaluations of AFs by exploiting decision procedures for problems of lower complexity whenever possible. Whether and how our general update semantics methodology can be applicable to the systematic study of algorithms for computing extensions, also has to be left to future research.

3.8 Conclusion and further research

In this chapter, we have proposed a methodologically novel approach to choosing extensions of argumentation frameworks by studying abstract and concrete commitment graphs that correspond to step-wise commitments about the choice of extension. Inspired by the principle-based approach to abstract argumentation, we have studied two principles that mappings from AFs to commitment graphs should satisfy.

We believe that there are many potential applications of our commitment graph methodology. We briefly sketch some of them.

Apart from the two types of concrete commitment graphs defined in this chapter, there are many other types of concrete commitment graphs that could be studied. For example, any algorithm for computing all extensions of a given AF with respect to a fixed semantics gives rise to a concrete commitment graph with respect to that semantics, namely by reducing the search tree of the algorithm to its commitment points, similarly as we have reduced the most exhaustive update to the fine-grained commitment graph. Studying the properties of these commitment trees could give novel insights into the study of algorithms for computing extensions.

Further principles of commitment graphs can be defined and studied. This will help to differentiate better between different semantics as well as the different commitment graphs that they give rise to.

Furthermore, one can study properties of the different commitment-paths (paths through the commitment graph) that a given commitment graph gives rise to. Here as well a principle-based approach can make sense: These principles could help to choose an extension in a systematic way, and could thus be very relevant to application of abstract argumentation in which a unique extension has to be chosen from the set of all extensions.

Though this is not our intention in this chapter, commitment graphs can also be used to generalize Dung's semantic framework, in the following sense. Instead of associating a set of extensions with a framework, we can associate a set of commitment graphs with a framework. These commitment graphs can be our most fine-grained commitment graphs, our SCC-recursive commitment graphs, or some other kind of commitment graphs. This idea is developed in more details in the next chapter.

In this chapter we have only studied the commitment graph methodology with respect to Dung's AFs, but the methodology could also be applied to extensions of Dung's formalisms such as bipolar AFs [48], ADFs [33], higher-order AFs [49] etc.

A further interesting line of future research is to study whether and how our methodology could be applied outside abstract argumentation, e.g. to structured argumentation, logic programming, answer set programming, Reiter's default logic, causal theories, social choice theory etc.

Chapter 4

Refining argumentation semantics

4.1 Introduction

Following the methodology in non-monotonic logic, logic programming and belief revision, formal argumentation theory defines a diversity of semantics. This diversity has the advantage that a user can select the semantics best fitting her application, but it leads also to various practical challenges. First of all, how to choose among the considerable number of semantics existing in the argumentation literature for a particular application? The behaviour of semantics on examples can already be insightful, and Baroni and Giacomin [50] address the need for more systematic comparison of semantics based on a set of principles. However, what to do when no currently considered semantics is perfect? May there be a better semantics that has not been discovered yet? How to guide the search for new and hopefully better argumentation semantics? In this chapter, we propose a new approach: the combination of abstract argumentation semantics. We focus on the following three research questions:

1. How to combine two abstract semantics to yield a third semantics?
2. In particular, how to obtain the complete semantics by combining the preferred and grounded semantics?
3. Can we meaningfully combine features of naive-based and complete-based semantics?

Concerning our first research question, there are various ways in which abstract argumentation semantics can be combined. For example, in multi-sorted argumentation [14, 15, 16], one part of the framework can be evaluated according to for example grounded semantics, whereas another part of the framework is evaluated according to the preferred semantics. Another approach manipulates directly the sets of extensions. For example, the grounded and preferred can be combined by simply returning both the grounded and preferred extensions. Both of these approaches have drawbacks. For multi-sorted argumentation, we need to specify explicitly which semantics must be applied to which part of the framework. For the direct combination method, the approach seems too coarse-grained and the number of ways to combine semantics seems relatively limited.

We therefore introduce a dynamic approach in this chapter which is based on the labeling approach to argumentation semantics, in which the three labels *in*, *out* and *undec* are used. In our dynamic approach, we define step-wise versions of standard semantics based on epistemic labelings, which associate with each argument a nonempty *set* of labels from $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$. Intuitively, the set represents uncertainty about the label. We start with labeling each argument of the framework with the set $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$. This represents that we do not know the labeling yet. Then in each step we refine the labels by removing some of the labels. Finally we end up with a single label for each argument, and thus with a standard labeling. To represent the possibility of multiple extensions, the steps are not deterministic. The steps are represented by an abstract *update relation*, which mathematically is simply a binary relation among epistemic labelings. There are many distinct update relations representing the same standard semantics, and it is this additional expressive power that we will use in our first approach to combining abstract argumentation semantics.

Concerning our second research question, it is well known that the grounded semantics outputs the smallest complete extension, and that the preferred semantics outputs maximal complete extensions [4]. This suggests that there is potential to recover all complete extensions using a mixture of the grounded and preferred semantics. There may be complete extensions that are neither minimal nor maximal, and that it is therefore non-trivial to recover all the complete extensions using the grounded and the preferred semantics, without loosing any complete extensions. Though the derivation of the complete semantics from the grounded and preferred semantics does not serve any practical purpose, it serves to show that our dynamic semantic framework has sufficient expressive power to combine abstract semantics. We therefore pursue this second question to showcase our combination operation.

We do not claim it to be possible to retrieve the full set of complete extensions from preferred and grounded extensions alone, as they do not provide sufficient information, even when represented as labelings. Indeed, some argumentation frameworks have the same preferred and the same grounded labelings, yet differ in their complete labelings. Let us examine a concrete example of two argumentation frameworks with the same preferred and grounded labelings, but different complete labelings.

Example 4.1.1. Consider the two AFs F_1 and F_2 depicted in Fig. 4.1. Both have $\{(a, \mathit{undec}), (b, \mathit{undec}), (c, \mathit{undec}), (d, \mathit{undec})\}$ as their grounded labeling, and $\{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{in}), (d, \mathit{out})\}$ and $\{(a, \mathit{out}), (b, \mathit{in}), (c, \mathit{out}), (d, \mathit{undec})\}$ as their preferred labelings. While these are also all the complete labelings for F_1, F_2 also has $\{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{undec}), (d, \mathit{undec})\}$ as a complete labeling which is neither preferred nor grounded. Hence, given nothing other than the preferred and grounded labelings of a framework, it is not feasible to always accurately retrieve the set of complete labelings.

Hence, the approaches we propose still take the structure of the framework into account when combining the different semantics.

Concerning the third research question, note that recently naive-based semantics like stage semantics [17] and CF2 semantics [18] have received some attention, for example in the work of Gaggl and Dvořák [19], who define a new semantics (*stage2*) that combines features of stage and CF2 semantics, and in the work of Cramer and Guillaume [20], who

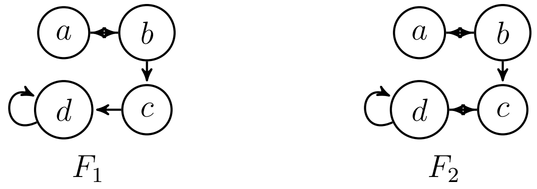


Figure 4.1: Two AFs with the same preferred and grounded labelings but different complete labelings.

performed an empirical study that showed that these naive-based semantics are better predictors of human argument acceptance than complete-based semantics like the grounded and preferred semantics.

For argumentation frameworks without odd cycles, the stage semantics fully agrees with the preferred semantics. One difference between the preferred semantics and the stage semantics is that the stage semantics generally provides a way to select accepted arguments even when odd cycles are around, whereas the preferred semantics tends to mark as *undecided* all arguments that are in an odd cycle or attacked by an odd cycle. One difference between the preferred semantics and the complete semantics is that the complete semantics allows one to locally not make choices for some unattacked even cycles while making choices for other unattacked even cycles, whereas in the preferred semantics one has to make choices for all unattacked even cycles. This motivates the following research question: Is there a sensible semantics that allows one to locally make choices for some unattacked odd or even cycles while not making choices for other unattacked odd or even cycles?

The layout of this chapter is as follows. After providing some preliminaries about argumentation semantics in Section 4.2, we introduces our dynamic approach based on epistemic labelings and update relations in Section 4.3. Section 4.4 addresses the second research question by showing how grounded and preferred semantics can be combined to obtain the complete semantics using an algorithmic approach to updates. As this approach is dependent on the choice of algorithm on which the update relation is based, we proceed in Section 4.5 to defining the *merge* of two argumentation semantics, a modification of our first approach that is applicable to any pair of semantics independently of any algorithmic considerations. In Subsection 4.5.1, we motivate the definition of the merge operator by considering how to use it to get the complete semantics from the grounded and preferred semantics without adding any algorithmic information. In Subsection 4.5.2, we show how the merge operator can be used to give rise to novel argumentation semantics, and, in particular, how it can be used to meaningfully combine features of naive-based and complete-based semantics. We conclude with an overview of further work in Section 4.6.

4.2 Preliminaries

An argumentation framework (AF) is a directed graph $\langle \mathcal{A}, \rightarrow \rangle$, where $\mathcal{A}$ is called the set of arguments, and $\rightarrow$ is called the attack relation. In this chapter, we do not consider enriched AFs such as bipolar AFs, EEAFs and weighted AFs [51].

Standard argumentation semantics come in two variants. Extension-based semantics associates with each AF a set of extensions (sets of the arguments). Labelling-based semantics attribute to each argument the label *in*, *out*or *undec*. The two approaches are interdefinable, in the sense that an argument is labeled *in* when it is in the extension, it is labeled *out* when it is not in the extension and there is an argument in the extension attacking it, and it is *undec* otherwise. Our dynamic approach uses an epistemic labelling, which associates with each argument a nonempty *set* of labels. Intuitively, the set represents uncertainty about the label.

We assume familiarity with 3-labeling semantics of argumentation frameworks as defined in [13]. Note that we will make use of the multi-labeling approach, where a set of labels is assigned to each argument. This set represents the possible labels for a given argument. The standard approach corresponds to the case where arguments are given singleton sets as labels.

We provide a reminder of labeling semantics below, but for more details we refer to the preliminaries, in particular Definition 2.1.5.

We define $\mathbb{L} = \{in, out, undec\}$ to be the set of possible *labels*.

Definition 4.2.1 (3-labeling)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF. We say that any function L from $\mathcal{A}$ to $\mathbb{L}$ is a 3-labeling of F.

The 3-labeling approach makes use of the notions of *legal labels*.

Definition 4.2.2 (Legal Labeling)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF, $a \in \mathcal{A}$ an argument and L a 3-labeling of F. We say that a is:

- *legally in* with respect to L iff $L(a) = in$ and for all $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) = out$;
- *legally out* with respect to L iff $L(a) = out$ and for some $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) = in$;
- *legally undecided* with respect to L iff $L(a) = undec$ and for all $b \in \mathcal{A}$ such that $(b, a) \in \rightarrow$, $L(b) \neq in$ and for at least one such b, $L(b) = undec$.

If all arguments in $\mathcal{A}$ are legally labeled with respect to L, then we say that L is a *complete labeling* of F. A complete labeling with a minimal set of *in*-labeled arguments is called a *grounded labeling*. A complete labeling with a maximal set of *in*-labeled arguments is called a *preferred labeling*. A complete labeling without *undec*-labeled arguments is called a *stable labeling*. A complete labeling with a minimal set of *undec*-labeled arguments is called a *semi-stable labeling*.

An *argumentation semantics* is a function that maps an argumentation framework to a set of labelings. The above defined notions give rise to the *complete*, *preferred* and *grounded* argumentation semantics. We call an argumentation semantics σ *complete-based* if all σ-labelings are complete labelings.

We will also refer to the *stage semantics* defined in its extension-based form by Verheij [17]. We adapt it to the labeling-based form by assigning the label *out* to all arguments that are not in the stage extension in Verheij's definition, even those which are not attacked by *in* arguments. This labeling-based form of the stage semantics can be defined as follows:

Definition 4.2.3 (Stage Labeling)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AF and L a 3-labeling of F. Define L_{in} to be the set $\{a \in \mathcal{A} \mid L(a) = \mathit{in}\}$. Define L^+_{in} to be the set $\{a \in \mathcal{A} \mid \exists b \in L_{\mathit{in}}.(b, a) \in \rightarrow\}$. We say that L is a *stage labeling* of F if L_{in} is conflict-free, $L_{\mathit{in}} \cup L^+_{\mathit{in}}$ is maximal with respect to set inclusion and $L(a) = \mathit{out}$ for all $a \in \mathcal{A} \setminus L_{\mathit{in}}$.

We also make use of the notions of *transitive closure* of a relation and *restriction* of a relation to a subset of its domain.

Definition 4.2.4 (Transitive Closure)**.** Let rel be a relation. We define the *transitive closure* of rel to be the smallest set rel^* such that $rel \subseteq rel^*$ and if $(a, b), (b, c) \in rel^*$, then $(a, c) \in rel^*$.

Definition 4.2.5 (Restriction)**.** Let A, B be sets, $A' \subseteq A$ and $R \subseteq A \times B$. We define the *restriction* of R to A' to be:

$$R \downarrow_{A'} = \begin{cases} \{(a, b) \in R \mid a, b \in A'\} & \text{if } A = B \\ \{(a, b) \in R \mid a \in A'\} & \text{otherwise} \end{cases}$$

The definition of restriction handles separately two cases: if the domain and range of the relation are the same, it then applies the restriction to both of them, for example in the case of the attack relation of an AF. In the case where the domain and range are different sets, it only performs the restriction on the domain set, for example in the case of a labeling function.

In the introduction, we have pointed out that retrieving the set of complete labelings from the preferred and grounded labelings alone is not feasible. We now provide a concrete example of two argumentation frameworks with the same preferred and grounded labelings, but different complete labelings.

4.3 Update relations

Standard labeling semantics provide a direct relation between an argumentation framework and a set of labeling functions, which attach to each argument exactly one label. We will now define update relations, which formalize the idea that the final labelings can be determined in a step-wise fashion. For this purpose, we introduce *epistemic labelings*, which associate with each argument a nonempty *set* of labels from $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$. The intuitive idea is that at a certain step in the update process, the set of labels associated with an argument tells us which labels we consider possible for this argument at this step. The steps in an update relation can be interpreted as moves in a dialogue, or as steps in an algorithm, or as learning a framework, or otherwise. Our dynamic semantic framework does not depend on such particular interpretations.

Notice that it makes little sense to separate the labeling function from the underlying framework, as the labeling is meaningless without it. We will hence consider pairs of argumentation framework and labeling functions.

Definition 4.3.1 (Labeled Argumentation Framework (LAF))**.** We define a *labeled argumentation framework* (LAF) to be a pair $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ where $\langle \mathcal{A}, \rightarrow \rangle$ is a finite argumentation framework and *EpLab* a function from $\mathcal{A}$ to $\mathcal{P}(\mathbb{L}) \setminus \{\emptyset\}$, called an *epistemic labeling*. Additionally, let $\mathbb{F}$ be the class of all labeled argumentation frameworks.

Observe that a labeling function cannot assign the empty set of labels to an argument, as the set of labels represents the possible final labels for that argument, and thus the empty set would mean that no label can be attached to it, which prevents us from having a final labeling for the framework.

Note that the notion of an epistemic labeling extends the notion of a partial extension from the previous chapter in the following way: when an argument is not covered by a partial extension, it is labelled $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$, $+a$ corresponds to the case where $\mathit{EpLab}(a) = \{\mathit{in}\}$ and $-a$ corresponds to $\mathit{EpLab}(a) = \{\mathit{out}, \mathit{undec}\}$. By having more expressivity in the labelings, we obtain more granularity in the extension graphs which provides us with more options for the combination of semantics.

We now introduce the notions of *initial* and *final* labeled frameworks, which correspond to the starting point and endpoint of a labeling process. In an initial LAF, every label is possible for each argument, while in a final LAF, every argument is assigned a singleton set of labels, representing the fact that a unique label has been selected.

Definition 4.3.2 (Initial and Final LAF)**.** Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ be a LAF. If for all $a \in \mathcal{A}$, $\mathit{EpLab}(a) \in \{\{\mathit{in}\}, \{\mathit{out}\}, \{\mathit{undec}\}\}$, we say that F is *final.* If for all $a \in \mathcal{A}$, $\mathit{EpLab}(a) = \mathbb{L}$, we say that F is *initial.*

Note that there is a one-to-one correspondence between the epistemic labelings *EpLab* of the final LAFs $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ and the 3-labelings of $\langle \mathcal{A}, \rightarrow \rangle$. This one-to-one correspondence can be formally defined by constructing singletons out of a given 3-labeling as follows:

Definition 4.3.3 ($T(\mathit{Lab})$)**.** Let $\langle \mathcal{A}, \rightarrow \rangle$ be an AF and *Lab* a 3-labeling of $\langle \mathcal{A}, \rightarrow \rangle$, define the epistemic labeling $T(\mathit{Lab})$ by $T(\mathit{Lab})(a) := \{\mathit{Lab}(a)\}$ for all $a \in \mathcal{A}$.

In this section with the basic definitions of our approach, we will be careful to make the formal distinction between a 3-labeling *Lab*, the corresponding epistemic labeling $T(\mathit{Lab})$ and the corresponding final LAF $(\langle \mathcal{A}, \rightarrow \rangle, T(\mathit{Lab}))$. In order to improve readability, we will not always make this distinction in later section, but instead identify the 3-labeling *Lab* with the corresponding epistemic labeling $T(\mathit{Lab})$ and the corresponding final LAF $(\langle \mathcal{A}, \rightarrow \rangle, T(\mathit{Lab}))$. For example, we might speak of an LAF being a complete labeling of a given argumentation framework, even though formally a complete labeling is a 3-labeling.

We now define a precision ordering on the LAFs based on the subset relation between the argument multi-labels, such that the final LAFs are the most precise and the initial LAFs are the least precise. Note however that only LAFs with the same underlying AF are comparable.

Definition 4.3.4 (Precision Ordering on LAFs)**.** Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ and $F' = (\langle \mathcal{A}', \rightarrow' \rangle, \mathit{EpLab}')$ be two labeled argumentation frameworks. We say that F is *at least as precise* as F' ($F \geq_p F'$), iff $\langle \mathcal{A}, \rightarrow \rangle = \langle \mathcal{A}', \rightarrow' \rangle$, and for all $a \in \mathcal{A}$, $\emptyset \subset \mathit{EpLab}(a) \subseteq \mathit{EpLab}'(a)$. We say that F is *more precise* than F' ($F >_p F'$) iff $F \geq_p F'$ and $F \neq_p F'$.

We will now define the central notion of this chapter, namely *update relations*, i.e. relations between LAFs which, starting from an initial LAF, monotonically increase precision, until a fixpoint is reached, at which point the LAF should be final and correspond to a desired output.

Definition 4.3.5 (Update Relation)**.** We say that *upd* $\subseteq \mathbb{F} \times \mathbb{F}$ is an *update relation* iff:

- for all $F' \in \mathbb{F}$ such that *upd*(F, F'), $F' \geq_p F$;
- if *upd*(F, F), then F is final.

Notice that by the definition of $\geq_p$, if F is final then $upd(F, F')$ implies $F = F'$.

We now define correspondence between update relations and direct semantics that formalizes the idea that an update relation can be viewed as a step-wise procedure that gives rise to a certain direct semantics. For this we first need an auxiliary definition.

Definition 4.3.6 (Reachable Fixpoint)**.** Let Rel be a relation on $\mathbb{F}$ and F an LAF. We say that F is *reachable* in Rel iff there exists an initial LAF F_i such that $(F_i, F) \in Rel^*$. We say that F is a *reachable fixpoint* in Rel iff F is reachable in Rel and $(F, F) \in Rel$.

Definition 4.3.7 (Giving Rise to a Semantics)**.** Let *upd* be an update relation and σ a semantics. We say that *upd gives rise* to σ iff for each 3-labeling *Lab* of $\langle \mathcal{A}, \rightarrow \rangle$, $(\langle A, R \rangle, T(\mathit{Lab}))$ is a reachable fixpoint in *upd* iff *Lab* is a σ labeling of $\langle \mathcal{A}, \rightarrow \rangle$.

The following theorem, which easily follows from Definition 4.3.5, provides a simple way of combining two given update relations to yield a third update relation:

Lemma 4.3.1. *If* upd_1 *and* upd_2 *are update relations, then* $\mathit{upd}_1 \cup \mathit{upd}_2$ *is an update relation.*

In Section 4.4 we will present an example where combining two update relations with a union operation gives us not only the union of the final labelings reachable by either of them, but also additional labelings. This means that the semantics that $\mathit{upd}_1 \cup \mathit{upd}_2$ gives rise to is not necessarily induced by the semantics that upd_1 and upd_2 separately give rise to.

We are now interested in the comparison of updates in terms of precision increase per step, i.e. in the granularity of update relations. The idea is that an update relation is more granular than another if it takes more steps to reach its final LAFs. First of all, notice that such a comparison only makes sense for updates which output the same final LAF, i.e. updates which give rise to the same semantics.

Definition 4.3.8 (Restriction to Relevant Paths ($\overline{\mathit{upd}}$))**.** Let *upd* be an update relation. We define the *restriction of upd to relevant paths* ($\overline{\mathit{upd}}$) to be the set of pairs in *upd* that are in some *upd*-path from an initial to a final LAF.

Definition 4.3.9 (Fine-Grained Ordering)**.** Let upd_1 and upd_2 be two update relations. We say that upd_1 is *at least as fined-grained* as upd_2 ($\mathit{upd}_1 \geq_g \mathit{upd}_2$) iff $\overline{\mathit{upd}_1}^* \supseteq \overline{\mathit{upd}_2}$.

We then abstractly define the *most fine-grained* update relation for a given labeling semantics.

Definition 4.3.10 (mfg_σ)**.** Let σ be a labeling semantics. We define mfg_σ to be the smallest update relation such that for all update relations *upd* that give rise to σ, we have $\mathit{mfg}_\sigma \geq_g$ *upd*.

Lemma 4.3.2. *For every standard semantics, there exists a unique* mfg_σ*.*

Proof. Define mfg_σ as follows: $(F, F') \in \mathit{mfg}_\sigma$ iff either $F = F'$ is a σ labeling, or the following three properties are satisfied:

- $F' >_p F$;
- $\nexists F''$ such that $F' >_p F'' >_p F$;
- there exists a final F_f which is a σ labeling such that $F_f \geq_p F'$.

By definition, mfg_σ includes all possible links in any relevant path from an initial to a final LAF which encompasses a σ labeling. Hence, for any update relation *upd* which gives rise to σ, $\overline{\mathit{mfg}_\sigma}^* \supseteq \overline{\mathit{upd}}$. Also, mfg_σ includes by definition only pairs which are on a relevant path, as the first alternative adds the endpoints of these paths and the third item of the second alternative ensures that the pairs are on a relevant path. The first and second items of the second alternative ensure also that a minimal amount of pairs are added, making mfg_σ as small as possible. Also, note that mfg_σ is well-defined since we only consider finite AFs, and thus $\geq_p$ is finite. □

In subsequent sections, we need the following notion of a sub-framework:

Definition 4.3.11 (Sub-framework)**.** Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ be a LAF and $S \subseteq \mathcal{A}$. We define the *sub-framework* of F generated by S to be $\mathit{Sub}(F, S) = (\langle S, \rightarrow\downarrow_S \rangle, \mathit{EpLab}\downarrow_S)$.

4.4 Case analysis: An algorithmic approach for combining preferred and grounded

In this section, we consider update relations which give rise to the preferred and grounded semantics, and which are motivated by algorithms for computing these semantics that have been described by Dauphin and Schulz [52].

The algorithmic update relation for the grounded semantics first identifies the arguments which are only being attacked by arguments which are already labeled $\{\mathit{out}\}$, labels them as $\{\mathit{in}\}$ and any argument they attack as $\{out\}$, and then repeats this process until no arguments can be further labeled, at which point it will label all remaining arguments as $\{\mathit{undec}\}$.

Definition 4.4.1 (Unattacked)**.** For any labeled argumentation framework $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$, we define the set of *unattacked arguments* to be $\mathit{unattacked}(F) = \{a \in \mathcal{A} \mid \mathit{EpLab}(a) \supsetneq \{\mathit{in}\} \wedge \forall b \in \mathcal{A}.((b, a) \in \rightarrow \rightarrow \mathit{EpLab}(b) = \{\mathit{out}\})\}$.

In an initial AF, the set of unattacked arguments will correspond to the arguments which do not have any attackers in the framework, while for final AFs, this set will be empty since it only considers arguments which are not finally labeled.

Definition 4.4.2 (*step_grnd*)**.** We define $\mathit{step\_grnd} \subseteq \mathbb{F} \times \mathbb{F}$ to be the relation such that $((\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}), (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}')) \in \mathit{step\_grnd}$ iff one of the following conditions holds:

- *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})) \neq \emptyset$, and $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}')$ is the least precise LAF that is more precise than $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ such that for all $a \in$ *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}))$, $\mathit{EpLab}'(a) = \{\mathit{in}\}$ and for all $c \in \mathcal{A}$ such that $(a, c) \in \rightarrow$ and $\mathit{out} \in \mathit{EpLab}(c)$, $\mathit{EpLab}'(c) = \{\mathit{out}\}$.

- *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})) = \emptyset$, there is an $a \in \mathcal{A}$ such that $\mathit{EpLab}(a) \supsetneq \{\mathit{undec}\}$, and $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}')$ is the least precise LAF that is more precise than $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ such that for all $a \in \mathcal{A}$ such that $\mathit{EpLab}(a) \supsetneq \{\mathit{undec}\}$, $\mathit{EpLab}'(a) = \{\mathit{undec}\}$.

- $(\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}) = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}')$ is a final LAF.

Note that before labeling arguments out, we ensure that it is a possibility, e.g. by having the condition $\mathit{out} \in \mathit{EpLab}(c)$ in the first item of Definition 4.4.2. While this requirement will straightforwardly be fulfilled in any reachable LAF, it is required to ensure that the increase in precision is satisfied even for those LAFs that are not reachable from an initial LAF.

The following lemma now easily follows from the above definition:

Lemma 4.4.1. *step_grnd is an update relation.*

The following theorem states that *step_grnd* does indeed have the intended property that it gives rise to the grounded labeling:

Theorem 4.4.2. *step_grnd gives rise to the grounded semantics.*

Proof sketch. One can easily see that whenever *step_grnd* changes the label of an argument a to $\{\mathit{in}\}, \{\mathit{out}\}$ or $\{\mathit{undec}\}$, argument a is legally labeled $\{\mathit{in}\}, \{\mathit{out}\}$ or $\{\mathit{undec}\}$ respectively. Thus the final labeling reachable in *step_grnd* is a complete labeling. To show that the final labeling reachable in *step_grnd* is the complete labeling that maximizes *undec*, suppose that there is some complete labeling *EpLab* of $\langle \mathcal{A}, \rightarrow \rangle$ and let $\mathcal{A}' = \{a \in \mathcal{A} \mid \mathit{EpLab}(a) = \{\mathit{undec}\}\}$. It is now enough to show that *step_grnd* never labels any $a \in \mathcal{A}'$ $\{\mathit{in}\}$ or $\{\mathit{out}\}$. Consider for a proof by contradiction the first step where *step_grnd* does label some $a \in \mathcal{A}'$ $\{\mathit{in}\}$, respectively $\{\mathit{out}\}$. Since a is legally labeled $\{\mathit{undec}\}$ in *EpLab*, some $a' \in \mathcal{A}'$ must attack a, so by Definitions 4.4.1 and 4.4.2, a' must already be labeled $\{\mathit{out}\}$ in a previous step, respectively there must exist an a' which has been labeled $\{\mathit{in}\}$ in a previous step, which is a contradiction. □

Let us now examine *step_pref*, a similar update relation which computes the preferred labelings. For this, we first define the notion of minimal non-trivial admissible sets of arguments, which resembles the notion of initial-like sets [36], but takes also the partial labels into account.

Definition 4.4.3 ($\mathit{min\_adm}(F)$)**.** Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ be a labeled argumentation framework. We define $\mathit{min\_adm}(F) \subseteq \mathcal{P}(\mathcal{A})$ to be the set of all minimal subsets S of $\mathcal{A}$ that satisfy the following conditions:

- $S \neq \emptyset$;

- for all $a \in S$, *EpLab*$(a) \supsetneq \{$*in*$\}$;
- for all $a, b \in S$, $(a, b) \notin \rightarrow$;
- for all $a \in S$ and $b \in \mathcal{A}$ such that *EpLab*$(b) \neq \{$*out*$\}$ and $(b, a) \in \rightarrow$, there exists $a' \in S$ such that $(a', b) \in \rightarrow$.

So the function $min\_adm(F)$ returns all minimal non-empty admissible sets of arguments whose label could still be changed to $\{$*in*$\}$. The update relation *step_pref* proceeds with a process similar to the one in the *step_grnd* update, iteratively labeling $\{$*in*$\}$ all arguments with all attackers $\{$*out*$\}$, and then labeling all arguments attacked by those as $\{$*out*$\}$. The difference lies in the case where *unattacked*(F) is empty, where the preferred update relation looks for minimal non-trivial admissible sets, label them $\{$*in*$\}$ and arguments they attack $\{$*out*$\}$.

Definition 4.4.4 (*step_pref*)**.** We define *step_pref* $\subseteq \mathbb{F} \times \mathbb{F}$ to be the relation such that $((\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}), (\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}')) \in$ *step_pref* iff one of the following conditions holds:

- *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})) \neq \emptyset$, and $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}')$ is the least precise LAF that is more precise than $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})$ such that for all $a \in$ *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}))$, *EpLab*$'(a) = \{$*in*$\}$ and for all $c \in \mathcal{A}$ such that $(a, c) \in \rightarrow$ and *out* $\in$ *EpLab*(c), *EpLab*$'(c) = \{$*out*$\}$.
- *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})) = \emptyset$, and for some $S \in min\_adm(F)$, $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}')$ is the least precise LAF that is more precise than $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})$ such that for all $a \in S$, *EpLab*$'(a) = \{$*in*$\}$ and for all $c \in \mathcal{A}$ such that $(a, c) \in \rightarrow$ and *out* $\in$ *EpLab*(c), *EpLab*$'(c) = \{$*out*$\}$.
- *unattacked*$((\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})) = min\_adm(F) = \emptyset$, and there is an $a \in \mathcal{A}$ such that *EpLab*$(a) \supsetneq \{$*undec*$\}$, and $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}')$ is the least precise LAF that is more precise than $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab})$ such that for all $a \in \mathcal{A}$ such that *EpLab*$(a) \supsetneq \{$*undec*$\}$, *EpLab*$'(a) = \{$*undec*$\}$.
- $(\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}) = (\langle \mathcal{A}, \rightarrow \rangle, \textit{EpLab}')$ is a final LAF.

The following lemma now easily follows from the above definition:

Lemma 4.4.3. *step_pref is an update relation.*

The following theorem, which can be proved in a similar way as Theorem 4.4.2, states that *step_pref* has the intended property that it gives rise to the preferred labeling:

Theorem 4.4.4. *step_pref gives rise to the preferred semantics.*

We now find the interesting result that combining these two update relations with a union operation gives us not only the union of the final labelings reachable by either of them, but also the complete labelings which are neither grounded nor preferred:

Theorem 4.4.5. *step_grnd* $\cup$ *step_pref gives rise to the complete semantics.*

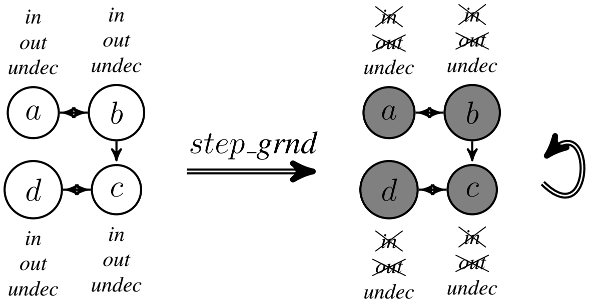


Figure 4.2: Example path from the initial LAF F to the corresponding final LAF in $step\_grnd$.

Proof sketch. One can easily see that any final labeling reachable in *step_grnd* $\cup$ *step_pref* is a complete labeling, as the two update relations preserve the legality of argument labels. So we only prove that each complete labeling *Lab* is reachable in *step_grnd* $\cup$ *step_pref*.

Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ be an initial LAF and EpLab_c the complete labeling we want to reach. First, apply either *step_grnd* or *step_pref* until we reach F' where the set of unattacked arguments is empty. At this point, the set S of *in* arguments is the grounded extension, and thus these arguments must also be *in* in EpLab_c, since the grounded extension is the intersection of all complete extensions (this follows from it being the unique smallest complete extension). Let S' be the set of arguments which are *in* in EpLab_c but not $\{\mathit{in}\}$ in F'. $S \cup S'$ forms an admissible set, since it is a complete extension. Hence, there is a minimal, non-empty subset of S', S'_1, such that $S \cup S'_1$ is admissible. There is an edge in the relation *step_pref* which labels the arguments in S'_1 as *in* and any argument they attack as *out*, according to Def. 4.4.4 second item. The rest of the arguments in S' are labeled *in* via Def. 4.4.4, either with the first item, or again with the second item as above. Once we have reached the LAF where all *in* arguments from EpLab_c are $\{\mathit{in}\}$ and any argument they attack $\{\mathit{out}\}$, we can make a step with *step_grnd* following Def. 4.4.2, second item, to label all remaining arguments as $\{\mathit{undec}\}$. We have then reached the fixpoint $F_f = (\langle \mathcal{A}, \rightarrow \rangle, T(\mathit{EpLab}_c))$, as desired.

□

Example 4.4.1. Let us examine the initial LAF $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ where $\mathcal{A} = \{a, b, c, d\}$, $\rightarrow = \{(a, b), (b, a), (b, c), (c, d), (d, c)\}$. Since $\mathit{unattacked}(F) = \emptyset$, *step_grnd* will send F to the fixpoint where all arguments are labeled $\{\mathit{undec}\}$. This is depicted in Fig. 4.2.

Let us now consider the same LAF F under the $\mathit{step\_pref}$ update relation this time. Again, $\mathit{unattacked}(F) = \emptyset$, but $\mathit{min\_adm}(F) = \{\{a\}, \{b\}, \{d\}\}$. The relation hence branches out in three paths. Let us focus the path with $\{a\}$. So the relation $\mathit{step\_pref}$ sends F to the LAF $F_{\mathit{pref}1}$ where a is $\{\mathit{in}\}$ and b is $\{\mathit{out}\}$, as depicted in Fig. 4.3. $\mathit{unattacked}(F_{\mathit{pref}1}) = \emptyset$, but $\mathit{min\_adm}(F_{\mathit{pref}1}) = \{\{c\}, \{d\}\}$, which gives us once again two possible directions in which to branch out. We will examine the one which selects $\{c\}$. This then gives us the final fixpoint $F_{\mathit{pref}2} = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}_{\mathit{pref}2})$, where $\mathit{EpLab}_{\mathit{pref}2}(a) = \mathit{EpLab}_{\mathit{pref}2}(c) = \{\mathit{in}\}$ and $\mathit{EpLab}_{\mathit{pref}1}(b) = \mathit{EpLab}_{\mathit{pref}1}(d) = \{\mathit{out}\}$.

We now consider the union of both relations. We can first send F to $F_{\mathit{pref}1}$ using the

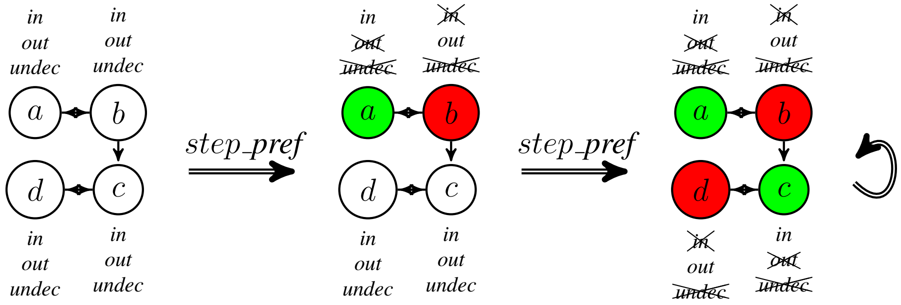


Figure 4.3: Example path from the initial LAF F to one of the corresponding final LAFs in *step_pref*.

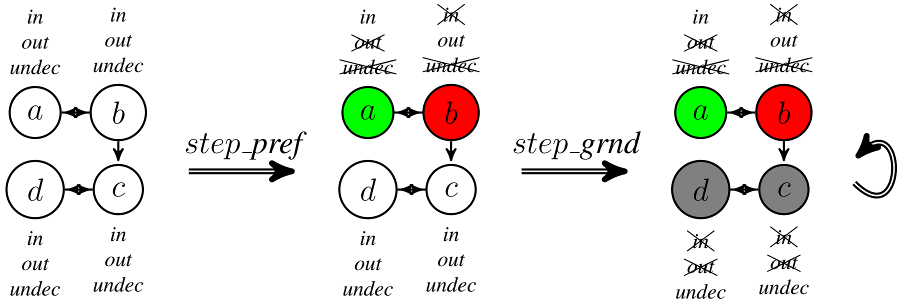


Figure 4.4: Example path from the initial LAF F to one of the corresponding final LAFs in *step_grnd* ∪ *step_pref* which neither update can reach by itself.

same step from *step_pref* as above. However this time we can apply *step_grnd* to F_{pref1}, and since $unattacked(F_{pref1}) = \emptyset$, the remaining arguments c and d are assigned the $\{undec\}$ label, sending F_{pref1} to the fixpoint F_{comp}, where a is $\{in\}$, b is $\{out\}$ and c, d are $\{undec\}$. Notice that F_{comp} corresponds to a complete labeling of F which is neither preferred nor grounded. This situation is depicted in Fig. 4.4.

4.5 Merging semantics through the most fined-grained update relation

In the previous section, we have shown that we can obtain the complete semantics by taking the union of two algorithmically motivated update relations giving rise to the grounded and the preferred semantics respectively. The success of this approach was dependent on the details of the algorithmic update relations that we defined, so it cannot be generalized to combine arbitrary semantics. In this section, we want to generalize our methodology to make it applicable to the combination of arbitrary semantics. For this purpose, we will examine a way to combine any two standard semantics via their most fine-grained update and a combination operation we call merging.

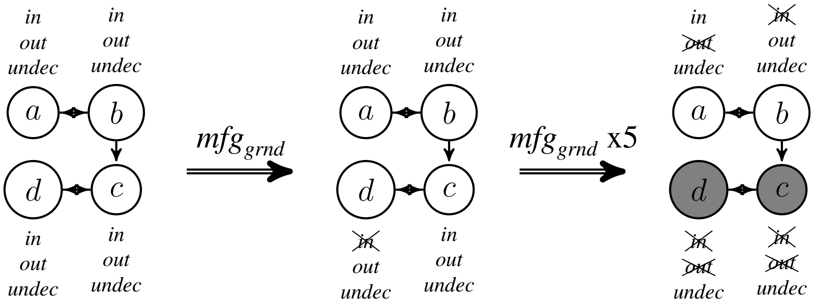


Figure 4.5: Example path from the initial LAF F to an intermediate LAF F' in $\mathit{mfg}_{\mathit{grnd}}$.

4.5.1 Merging preferred and grounded

If we were to attempt to combine $\mathit{mfg}_{\mathit{pref}}$ and $\mathit{mfg}_{\mathit{grnd}}$ by simply taking their union, as we have done in the algorithmic approach, it follows from their definition that we would simply obtain as reachable fixpoints the labelings which are either preferred or grounded. The main issue is that $\mathit{mfg}_{\mathit{pref}}$ and $\mathit{mfg}_{\mathit{grnd}}$ are not applicable to LAFs which do not agree with some final LAF of that semantics.

For an example of this issue, we consider again the same LAF F as in Example 4.4.1. Suppose we want to reach the same complete labeling that we reached in Figure 4.4, i.e. the one in which a is $\{\mathit{in}\}$, b is $\{\mathit{out}\}$, and c and d are $\{\mathit{undec}\}$. We could start by doing those six steps of $\mathit{mfg}_{\mathit{grnd}}$ that are compatible with the complete labeling that we want to reach, as depicted in Figure 4.5, yielding the intermediate LAF F'. Now we would like to apply $\mathit{mfg}_{\mathit{pref}}$ to F' in order to delete the *undec*-labels from a and b. However, $\mathit{mfg}_{\mathit{pref}}$ cannot be applied at all to F', as F' is not compatible with any preferred labeling of F.

So instead of just taking the union of $\mathit{mfg}_{\mathit{pref}}$ and $\mathit{mfg}_{\mathit{grnd}}$, we will define a more complicated operation called the *merge* of two update relations, which we denote by $\mathit{upd}_1 \uplus \mathit{upd}_2$. The idea is that once neither $\mathit{mfg}_{\mathit{pref}}$ nor $\mathit{mfg}_{\mathit{grnd}}$ allow us to get closer to a desired complete labeling, we will focus on a particular sub-framework and draw analogies with another framework which also contains that sub-framework. This operation resembles the way input is imposed in multi-sorted argumentation semantics [53]. The details of this approach are somewhat complicated, so let us first sketch the approach by seeing how it can be applied to the example that we just looked at.

The idea is that we focus on the set $S = \{a, b\}$ of arguments, as we want to remove labels from a and b. In order to work with $\mathit{mfg}_{\mathit{pref}}$ on the sub-framework $\mathit{Sub}(F', S)$ induced by S, we consider an alternative framework F_2 that also has $\mathit{Sub}(F', S)$ as a sub-framework, but to which $\mathit{mfg}_{\mathit{pref}}$ can be applied. A suitable choice of F_2 is depicted on the left in Figure 4.6. Now we apply $\mathit{mfg}_{\mathit{pref}}$ twice to F_2 as depicted in Figure 4.6, removing the labels from a and b that we wanted to remove. If certain conditions are satisfied, we may import the changes we have made to F_2 back to F, as depicted in Figure 4.7.

Now what are the conditions that need to be satisfied in order to allow for this import of changes from one framework to another? In order to describe these conditions, we need to split the original framework into three parts, based on sets of arguments:

- S, the arguments we will focus on;

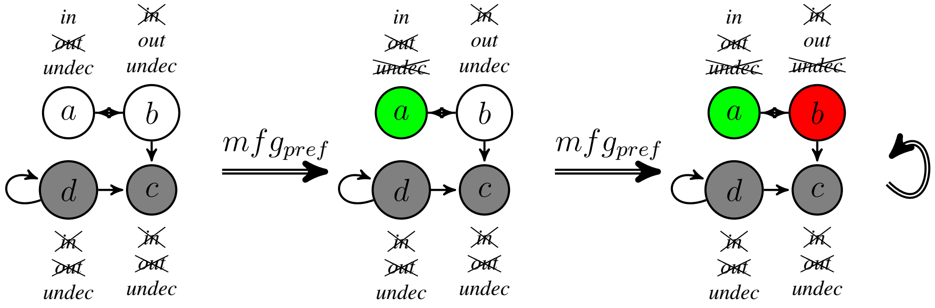


Figure 4.6: Example path on a parallel F_2 framework with $S = \{a, b\}$ and $I = \{c\}$, where mfg_{pref} is applicable.

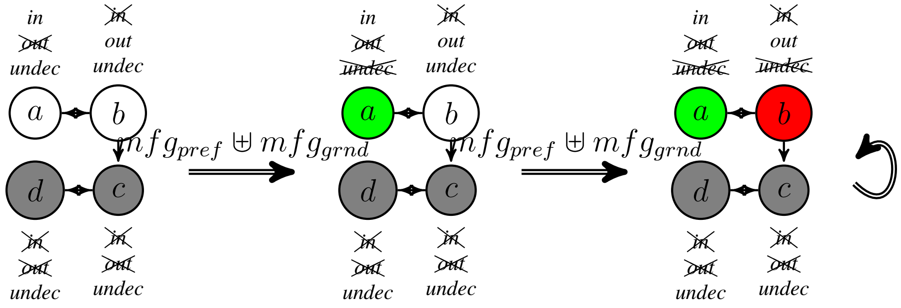


Figure 4.7: Importing the steps made in Fig. 4.6 into F' allows us to reach a complete labeling which is neither grounded nor preferred.

- I, called the *interface*, which is a set of arguments which already have a maximally precise label (i.e. a singleton) and which separate the set S from the rest of the framework;
- $\mathcal{A} \setminus (S \cup I)$, the rest of the framework, on which the two frameworks may differ.

The basic idea is that in order to import some change that an update relation mfg_σ can make on F_2 to the LAF F, we have to choose F_2 in such a way that in both F and F_2, the interface I separates S from the rest of the framework. Furthermore, we have to choose F_2 in such a way that mfg_σ can actually be applied to F_2, which is only possible if the maximally precise labels of the arguments in I are possible labels for these arguments in F_2 under the semantics σ.

We are now ready to present the formal definition of the merge $\mathit{upd}_1 \uplus \mathit{upd}_2$:

Definition 4.5.1 (Merge)**.** Let upd_1 and upd_2 be two update relations. We define the *merge* of upd_1 and upd_2 ($\mathit{upd}_1 \uplus \mathit{upd}_2$) as the smallest relation such that:

1. $\mathit{upd}_1 \uplus \mathit{upd}_2 \supseteq \mathit{upd}_1 \cup \mathit{upd}_2$;
2. For $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ and $F' = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}')$, $(F, F') \in \mathit{upd}_1 \uplus \mathit{upd}_2$ if there exist disjoint sets $S, I \subseteq \mathcal{A}$ and two LAFs $F_2 = (\langle \mathcal{A}_2, \rightarrow_2 \rangle, \mathit{EpLab}_2)$ and $F_2' = (\langle \mathcal{A}_2, \rightarrow_2 \rangle, \mathit{EpLab}_2')$ such that the following conditions are satisfied:

   a. $(F_2, F_2') \in \mathit{upd}_1 \cup \mathit{upd}_2$;

   b. $\mathit{Sub}(F_2, S \cup I) = \mathit{Sub}(F, S \cup I)$;

   c. $\forall s \in S, \forall a \in \mathcal{A} \setminus (I \cup S), (s, a), (a, s) \notin \rightarrow, \rightarrow_2$;

   d. $\forall a \in I$, $\mathit{EpLab}(a) = \mathit{EpLab}_2(a)$ is a singleton;

   e. $\mathit{EpLab}_2' \downarrow_S \neq \mathit{EpLab}_2 \downarrow_S$;

   f. $\mathit{EpLab}_2' \downarrow_{\mathcal{A}_2 \setminus S} = \mathit{EpLab}_2 \downarrow_{\mathcal{A}_2 \setminus S}$;

   g. $\mathit{Sub}(F, \mathcal{A} \setminus S)$ is reachable by $\mathit{upd}_1 \uplus \mathit{upd}_2$;

   h. $\mathit{EpLab}' \downarrow_S = \mathit{EpLab}_2' \downarrow_S$.
3. if F is final and reachable by $\mathit{upd}_1 \uplus \mathit{upd}_2$, then $(F, F) \in \mathit{upd}_1 \uplus \mathit{upd}_2$.

Given the complexity of this definition, let us explain it a bit more: Item 1 expresses the fact that we can still perform any step which is available in either one of the base updates. However, as we have seen previously, this is not enough in order to obtain meaningful combinations of most fine-grained updates, which is why we have item 2. Given a labeled argumentation framework F, additional changes are potentially possible if we can identify two disjoint sets of arguments S and I, where S is the set of arguments we are interested in and where the update will be occurring and I is a fully-labeled interface between S and the rest of the framework, meaning that no argument in S attacks nor is attacked by an argument in $\mathcal{A} \setminus (S \cup I)$. Once such sets have been identified, we observe other labeled argumentation frameworks F_2 which also contain $S \cup I$ with the same structure and epistemic labels but can differ in structure and labels in the rest of the framework. If an update with $\mathit{upd}_1 \cup \mathit{upd}_2$

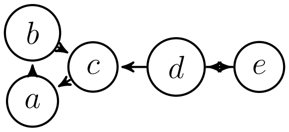


Figure 4.8: Example AF F to illustrate the need for item g) in Def. 4.5.1

is possible in such a framework, we then allow this change to be imported into F to produce F'. In more details, sub-item *a* specifies that there must be a $upd_1 \cup upd_2$ step which relates F_2 to F_2'. Sub-item *b* ensures that the parallel framework F_2 agrees with F on the structure and epistemic labels of $S \cup I$. Sub-item *c* guarantees that there are no connections between S and $\mathcal{A} \setminus (S \cup I)$ in neither F nor F_2. Sub-item *d* ensures that I is fully labeled, which is required in order to ensure the well-behavior of the merge operation. The idea is that once this interface has been fully labeled by one of the two updates, if we can modify $\mathcal{A} \setminus (S \cup I)$ in order to make sense of these labels for the second update, then we can also perform steps from this second update inside S, and then by perhaps modifying $\mathcal{A} \setminus (S \cup I)$ again we can switch back to using the first update again and so on. Sub-item *e* ensures that change happens inside S, while sub-item *f* ensures that no change is made outside of S, so that change happens in S and exclusively there. Sub-item *g* provides an additional restriction on the partitioning to ensure that for an argument i in the interface I which has a justification $a \in S$ for its label which hasn't been assigned yet, we do not introduce a new justification in $\mathcal{A} \setminus (S \cup I)$ for i's label and hence allow for a different label to be assigned to a, leaving i with no justification for its label in F'. This is clarified in Example 4.5.1. Sub-item *h* simply specifies that the change made in the parallel LAF be imported into the original one to produce F', and combined with the sub-item *e* entails that a change within S is necessary between F_2 and F_2'. Finally, with item 3 we ensure that reachable final frameworks are also fixpoints, which is needed since the second item of the definition does not produce any fixpoints, as it requires some change to happen in the LAF with the first sub-item.

Example 4.5.1. Consider the AF F depicted in Fig. 4.8. Since $\{(a, \textit{in}), (b, \textit{out}), (c, \textit{out}), (d, \textit{in}), (e, \textit{out})\}$ is a preferred labeling, it is possible to assign the *out* label to c via $mfg_{\textit{pref}}$, as well as the *in* label to a and the *out* label to b. From there, it would be possible to set $I = \{c\}$ and $S = \{d, e\}$, allowing one to import changes from the parallel framework F' depicted in Fig. 4.9. Here, a few steps in $mfg_{\textit{grnd}}$ would assign the *undec* label to d and e. This would however produce a labeling where c is *out*, but has no reason to be labelled so, since d is *undec*. This kind of scenario is prevented by item g) of Def. 4.5.1 as no sub-LAF consisting of $\{a, b, c\}$ where c is *out* is reachable with $mfg_{\textit{pref}}$ or $mfg_{\textit{grnd}}$. Thus, item g) forces what we informally refer to as a justification for the interface's label to be either part of it, or contained in S.

In definition 4.5.1, we have defined the merge between two arbitrary update relations. In this chapter, we always apply this merge operation to two maximally fine-grained update relations and focus on the semantics that the resulting update relation gives rise to. In this way, the notion of a merge between two update relations gives rise to the following notion of a merge between two argumentation semantics:

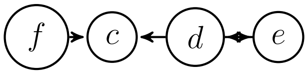


Figure 4.9: Argumentation framework F' parallel to F from Fig. 4.8. Here c is *out* due to f, allowing mfg_{grnd} to assign the *undec* label to both d and e.

Definition 4.5.2 ($\uplus$)**.** Given two argumentation semantics σ_1 and σ_2, we define $\sigma_1 \uplus \sigma_2$ to be the semantics that $\mathit{mfg}_{\sigma_1} \uplus \mathit{mfg}_{\sigma_2}$ gives rise to.

We originally motivated the definition of the merge operation with the goal to combine the grounded and preferred semantics to yield the complete semantics. The following theorem establishes that this is indeed the case for the merge operation as we have defined it.

Theorem 4.5.1. *preferred* $\uplus$ *grounded* $=$ *complete.*

Proof. By Definition 4.5.2, we need to show that $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ gives rise to the complete semantics. So we need to prove that every complete labeling is a reachable fixpoint in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ and that every labeling that is a reachable fixpoint in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ is a complete labeling. We start by proving that every complete labeling is a reachable fixpoint in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$:

Let $\langle \mathcal{A}, \rightarrow \rangle$ be an argumentation framework, and let *Lab* be the complete labeling of $\langle \mathcal{A}, \rightarrow \rangle$ we want to reach. We want to show that $F_f = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}_f)$ is a reachable fixpoint, where $\mathit{EpLab}_f = T(\mathit{Lab})$. Let $C = \{a \in \mathcal{A} \mid \mathit{Lab}(a) = \mathit{in}\}$, $I = \{b \in \mathcal{A} \mid \mathit{Lab}(a) = \mathit{out}\}$ and $S = \mathcal{A} \setminus (C \cup I)$. Consider the LAF $F_i = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}_i)$, where for all $a \in C$, $\mathit{EpLab}_i(a) := \{in\}$, for all $b \in I$, $\mathit{EpLab}_i(b) := \{out\}$, and for all $c \in S$, $\mathit{EpLab}_i(c) := \{\mathit{in}, \mathit{out}, \mathit{undec}\}$. Since *Lab* is a complete labeling, C is admissible, so there exists a preferred labeling where all arguments in C are *in*. Thus F_i is reachable with $\mathit{mfg}_{\mathit{pref}}$.

We now want to apply item 2 of Definition 4.5.1 to F_i multiple times in order to remove all the *in* and *out* labels from the arguments in S. For this purpose, we choose a "new" argument z, i.e. an argument $z \notin \mathcal{A}$, and consider the LAF $F_2 = (\langle \mathcal{A}_2, \rightarrow_2 \rangle, \mathit{EpLab}_2)$ where $\mathcal{A}_2 = (\mathcal{A} \setminus C) \cup \{z\}$, $\rightarrow_2 = \rightarrow\downarrow_{\mathcal{A}_2} \cup \{(z, a) \mid a \in I\}$ and $\mathit{EpLab}_2 = \mathit{EpLab} \downarrow_{\mathcal{A}_2} \cup \{(z, \{in\})\}$. Consider the final LAF $F_{2f} = (\langle \mathcal{A}_2, \rightarrow_2 \rangle, \mathit{EpLab}_{2f})$, where for all $a \in \mathcal{A}_2 \setminus S$, $\mathit{EpLab}_{2f}(a) = \mathit{EpLab}_2(a)$ and for all $a \in S$, $\mathit{EpLab}_{2f}(a) = \{\mathit{undec}\}$.

We want to show that F_{2f} is grounded. For this purpose, we first establish that F_{2f} is complete, i.e. that all labels in F_{2f} are legal labels: z is unattacked and is therefore legally labeled *in* in F_{2f}. All arguments in I are attacked by z, so they are legally labeled *out* in F_{2f}. Furthermore, since C does not defend any arguments it does not contain, every argument in S is attacked by at least one other argument in S. Additionally, the only *in* argument, z, does not attack any arguments in S. Thus the arguments in S are legally labeled *undec* in F_{2f}. Therefore, F_{2f} is a complete LAF, and since the only *in* argument, z, has to be labeled *in*, it is also grounded.

Therefore F_{2f} is reachable in $\mathit{mfg}_{\mathit{grnd}}$ from F_2. So by multiple applications of $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$, using item 2 of Def 4.5.1, one can reach F_f from F_i. Since F_f is final, F_f is a fixpoint, and thus F_f is a reachable fixpoint.

So far, we have shown that every complete labeling is a reachable fixpoint in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$. Now we still need to show that every labeling that is a reachable fixpoint in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ is a complete labeling:

Let $F = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab})$ be a reachable LAF in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$. We show by induction on $|\mathcal{A}|$ that there exists a final complete LAF which is at least as precise as F.

Induction hypothesis 1: Assume that for every LAF $F' = (\langle \mathcal{A}', \rightarrow' \rangle, \mathit{EpLab}')$ such that $|\mathcal{A}'| < |\mathcal{A}|$ and F' is reachable in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$, there exists a final complete LAF which is at least as precise as F'.

We now use a second induction on the steps required to reach F.

Base case: F is initial. Since there always exists a complete labeling for any framework, there exists a final complete LAF more precise than F.

Inductive step: F is not initial, but is reached in $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ through an LAF $F^* \neq F$ for which the required property holds. In other words, we have the following induction hypothesis for F^*:

Induction hypothesis 2: Assume that for $F^* = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}^*)$ such that $F^* \neq F$ and $(F^*, F) \in \mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$, there exists a final complete LAF $F^*_f = (\langle A, R \rangle, \mathit{EpLab}^*_f)$ such that $F^* \leq_p F^*_f$.

We distinguish three cases:

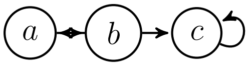
1. $(F^*, F) \in \mathit{mfg}_{\mathit{pref}}$. Then, by the definition of $\mathit{mfg}_{\mathit{pref}}$, there exists a final LAF which represents a preferred labeling of $\langle \mathcal{A}, \rightarrow \rangle$ and is at least as precise as F. Since preferred labelings are also complete, we are done.

2. $(F^*, F) \in \mathit{mfg}_{\mathit{grnd}}$. Similarly to the case above, it follows from the definition of $\mathit{mfg}_{\mathit{grnd}}$ that there exists a complete final LAF which is at least as precise as F.

3. $(F^*, F) \notin \mathit{mfg}_{\mathit{pref}} \cup \mathit{mfg}_{\mathit{grnd}}$. Since $F^* \neq F$, the item 2 of Definition 4.5.1 must be satisfied. In other words, there exist disjoint sets $S, I \subseteq \mathcal{A}$ and two LAFs F_2 and F'_2 that satisfy the conditions *a* to *h* from item 2 of Definition 4.5.1. By condition *a*, $(F_2, F'_2) \in \mathit{mfg}_{\mathit{pref}} \cup \mathit{mfg}_{\mathit{grnd}}$, so by the same reasoning as in cases 1 and 2 above, we can conclude that there exists a final LAF $F_{f2} = (\langle \mathcal{A}_2, \rightarrow_2 \rangle, \mathit{EpLab}_{f2})$ which is complete and at least as precise as F'_2. Also, by condition *g*, $F_s = \mathit{Sub}(F^*, \mathcal{A} \setminus S)$ is reachable by $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$, and by condition *e*, $S \neq \emptyset$, i.e. $|\mathcal{A} \setminus S| < |\mathcal{A}|$. So by induction hypothesis 1, there exists a final complete LAF $F_{sf} = (\langle \mathcal{A} \setminus S, \rightarrow\downarrow_{\mathcal{A} \setminus S} \rangle, \mathit{EpLab}_{sf})$ such that $F_{sf} \geq_p F_s$.

   We now construct the final LAF $F_f = (\langle \mathcal{A}, \rightarrow \rangle, \mathit{EpLab}_f)$ as follows: For all $a \in \mathcal{A} \setminus S$, $\mathit{EpLab}_f(a) := \mathit{EpLab}_{sf}(a)$, and for all $a \in S$, $\mathit{EpLab}_f(a) = \mathit{EpLab}_{f2}(a)$. From the construction of F_f and conditions *f* and *h* of Definition 4.5.1, it follows that F_f is more precise than F. To complete the proof, we now still need to show that F_f is a complete labeling, i.e. that all arguments in $\mathcal{A}$ are legally labeled in F_f.

   According to the definition of legal labeling (Definition 4.2.2), a labeling being legal depends only on the label of the arguments it is directly attacking or attacked by. According to condition *c* of Definition 4.5.1, the only arguments which are attacking or attacked by arguments in S are in $S \cup I$. The arguments in S are legally labeled in



F_{2f}, and thus they are also legally labeled in F_f, since $\mathit{Sub}(F_{2f}, S \cup I) = \mathit{Sub}(F_f, S \cup I)$. Similarly, since the arguments in $A \setminus (S \cup I)$ are legally labeled in F_{sf}, they are also legally labeled in F_f. Now take an arbitrary $a \in I$. We distinguish three cases:

(a) $\mathit{EpLab}_f(a) = \{\mathit{out}\}$: Then, since F_{sf} is complete, there exists an argument $b \in \mathcal{A} \setminus S$ such that $(b, a) \in \rightarrow$ and $\mathit{EpLab}_f(b) = \{\mathit{in}\}$. So a is legally *out* in F_f.

(b) $\mathit{EpLab}_f(a) = \{\mathit{in}\}$: Then, since F_{sf} is complete, for all $b \in \mathcal{A} \setminus S$ such that $(b, a) \in \rightarrow$, $\mathit{EpLab}_f(b) = \{\mathit{out}\}$. Also, since F_{2f} is complete, for all $b \in S$ such that $(b, a) \in \rightarrow$, $\mathit{EpLab}_f(b) = \{\mathit{out}\}$. Hence, a is legally *in* in F_f.

(c) $\mathit{EpLab}_f(a) = \{\mathit{undec}\}$: Then, since since F_{sf} is complete, for all $b \in \mathcal{A} \setminus S$ such that $(b, a) \in \rightarrow$, $\mathit{EpLab}_f(b) \neq \{\mathit{in}\}$, and for at least one such b, $\mathit{EpLab}_f(b) = \{\mathit{undec}\}$. Also, since F_{2f} is complete, for all $b \in S$ such that $(b, a) \in \rightarrow$, $\mathit{EpLab}_f(b) \neq \{\mathit{in}\}$. Hence, a is legally *undec* in F_f.

So all arguments in F_f are legally labeled and thus F_f is complete. Hence, there exists a final complete LAF which is at least as precise as F.

Therefore, for all reachable LAFs, there exists a complete LAF which is at least as precise. Since $\mathit{mfg}_{\mathit{pref}} \uplus \mathit{mfg}_{\mathit{grnd}}$ is an update relation, every reachable fixpoint is final, and thus every reachable fixpoint is complete.

□

4.5.2 Defining new semantics via merging

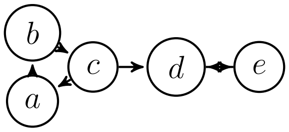
The merge operation defined in Definition 4.5.2 can be used to combine two arbitrary argumentation semantics to yield another argumentation semantics. So far, we have shown that merging grounded and preferred semantics yields the complete semantics. In this section, we show how applying this merge operation to other pairs of semantics gives rise to completely new argumentation semantics.

First, notice that the second part of the proof of Theorem 4.5.1 only makes use of the fact that the labelings reached at the different stages are complete, but not of any other properties particular to preferred or grounded. Hence, the merge of any two complete-based semantics is a complete-based semantics itself, i.e. all the labelings it returns are also complete.

Theorem 4.5.2. *Let σ_1 and σ_2 be two complete-based argumentation semantics. Then $\sigma_1 \uplus \sigma_2$ is also a complete-based semantics.*

For example, by merging stable and grounded, we obtain labelings which are complete. However, in this case, we do not recover all complete labelings as we did when merging grounded and preferred. Let us examine this short example to see why.

Example 4.5.2. Consider the following argumentation framework F:



Using $\mathit{mfg}_{\mathit{stab}} \uplus \mathit{mfg}_{\mathit{grnd}}$, one can reach the labelings $\{(a, \mathit{out}), (b, \mathit{in}), (c, \mathit{out})\}$ and $\{(a, \mathit{undec}), (b, \mathit{undec}), (c, \mathit{undec})\}$. However, suppose we wish to reach the complete labeling $\mathit{EpLab} = \{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{undec})\}$. Since there is no stable labeling, we cannot make any steps via $\mathit{mfg}_{\mathit{stab}}$ from the initial LAF. Also, attempting to find a similar framework from which one could import changes, will not work at this point where the LAF is initial, because the interface I would have to be empty, which only works for disconnected AFs.

Hence, one can only make steps in $\mathit{mfg}_{\mathit{grnd}}$ in order to reduce c's epistemic labeling to $\{undec\}$, a's to $\{in, undec\}$ and b's to $\{out, undec\}$. This, however, is as close as one can get to *EpLab* using $\mathit{mfg}_{\mathit{stab}} \uplus \mathit{mfg}_{\mathit{grnd}}$. Any F_2 satisfying the conditions of item 2 of Definition 4.5.1 must have $I = \{c\}$. In this case, the set S on which we want to make changes would have to be $\{a, b\}$. But then $I \cup S$ includes all arguments, so that F_2 would have to be identical to F, so that we cannot use item 2 of Definition 4.5.1 to make any change that we cannot already make with item 1 of Definition 4.5.1.

Therefore, *EpLab* is unreachable with $\mathit{mfg}_{\mathit{stab}} \uplus \mathit{mfg}_{\mathit{grnd}}$.

An interesting note to make is that all labelings reachable by $\mathit{mfg}_{\mathit{stab}} \uplus \mathit{mfg}_{\mathit{grnd}}$ are complete, according to Theorem 4.5.2, and hence this combination provides a novel complete-based semantics which returns more labelings than both the stable semantics and the grounded semantics.

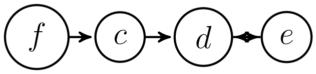
Similarly, the merge of the semi-stable and grounded semantics returns a novel complete-based semantics. One can check this by replacing stable by semi-stable in the situation described in Example 4.5.2: The desired complete labeling is still unreachable.

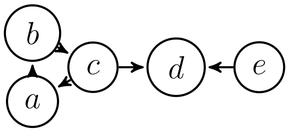
As motivated in the introduction, we are also interested in the following research question related to combining features of naive-based and complete-based semantics: Is there a sensible semantics that allows one to locally make choices for some unattacked odd or even cycles while not making choices for other unattacked odd or even cycles. Given our methodology for merging semantics, an obvious candidate for such a semantics is *stage* $\uplus$ *grounded*, i.e. the semantics resulting from merging the stage semantics with the grounded semantics. By considering its application to an example, we show that this semantics does indeed have this feature.

Example 4.5.3. Consider the following argumentation framework F':

The stage labelings of F' are

$$\begin{aligned}
\mathit{EpLab}_1 &= \{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{out}), (d, \mathit{in}), (e, \mathit{out})\},\\
\mathit{EpLab}_2 &= \{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{out}), (d, \mathit{out}), (e, \mathit{in})\},\\
\mathit{EpLab}_3 &= \{(a, \mathit{out}), (b, \mathit{in}), (c, \mathit{out}), (d, \mathit{in}), (e, \mathit{out})\},\\
\mathit{EpLab}_4 &= \{(a, \mathit{out}), (b, \mathit{in}), (c, \mathit{out}), (d, \mathit{out}), (e, \mathit{in})\},\\
\mathit{EpLab}_5 &= \{(a, \mathit{out}), (b, \mathit{out}), (c, \mathit{in}), (d, \mathit{out}), (e, \mathit{in})\}.
\end{aligned}$$





Its grounded labeling is $\mathit{EpLab}_6 = \{(a, \mathit{undec}), (b, \mathit{undec}), (c, \mathit{undec}), (d, \mathit{undec}), (e, \mathit{undec})\}$. Additionally to these six labelings, it has three further *stage* ⊎ *grounded*-labelings:

$$\begin{aligned}
\mathit{EpLab}_7 &= \{(a, \mathit{in}), (b, \mathit{out}), (c, \mathit{out}), (d, \mathit{undec}), (e, \mathit{undec})\},\\
\mathit{EpLab}_8 &= \{(a, \mathit{out}), (b, \mathit{in}), (c, \mathit{out}), (d, \mathit{undec}), (e, \mathit{undec})\},\\
\mathit{EpLab}_9 &= \{(a, \mathit{undec}), (b, \mathit{undec}), (c, \mathit{undec}), (d, \mathit{out}), (e, \mathit{in})\}.
\end{aligned}$$

EpLab_7 can be reached using $\mathit{mfg}_{\mathit{stage}} \uplus \mathit{mfg}_{\mathit{grnd}}$ by first applying $\mathit{mfg}_{\mathit{stage}}$ several times to reduce the epistemic labels on a, b and c to $\{\mathit{in}\}$, $\{\mathit{out}\}$ and $\{\mathit{out}\}$ respectively and then applying item 2 of Definition 4.5.1 with the interface $I := \{c\}$, the set $S := \{d, e\}$ and the following parallel framework F_2^7:

We can then apply $\mathit{mfg}_{\mathit{grnd}}$ multiple times to this parallel framework to reduce the epistemic labels of d and e to $\{\mathit{undec}\}$ and import these changes to the labeling on the main framework F' using item 2 of Definition 4.5.1. EpLab_8 can be reached using $\mathit{mfg}_{\mathit{stage}} \uplus \mathit{mfg}_{\mathit{grnd}}$ in a similar way using the same parallel framework.

EpLab_9 can be reached using $\mathit{mfg}_{\mathit{stage}} \uplus \mathit{mfg}_{\mathit{grnd}}$ by first applying $\mathit{mfg}_{\mathit{stage}}$ several times to reduce the epistemic labels on d and e to $\{\mathit{out}\}$ and $\{\mathit{in}\}$ respectively and then applying item 2 of Definition 4.5.1 with the interface $I := \{d\}$, the set $S := \{a, b, c\}$ and the following parallel framework F_2^9:

We can then apply $\mathit{mfg}_{\mathit{grnd}}$ multiple times to this parallel framework to reduce the epistemic labels of a, b and c to $\{\mathit{undec}\}$ and import these changes to the labeling on the main frameowrk F' using item 2 of Definition 4.5.1.

The stage semantics forces us to make a choice on the odd cycle $\{a, b, c\}$, and unless we choose to accept the argument c that attacks the even cycle, we are also forced to make a choice on the even cycle $\{d, e\}$. In the grounded semantics, there are no choices and all arguments become undecided. In *stage* ⊎ *grounded*, we can combine these features of stage and grounded: We can for example choose a from the odd cycle, but stay undecided about the arguments in the even cycle – this possible choice is formalized by EpLab_7.

So *stage* ⊎ *grounded* allows one to locally make choices for some unattacked odd or even cycles while not making choices for other unattacked odd or even cycles. It thus provides a positive answer to our third research question from the introduction.

4.6 Conclusion and future work

In this chapter we introduce a dynamic approach to combine two argumentation semantics to yield a third one. In particular, we provide a formal environment for the analysis of stepwise relations between labeled framework with an increase in the label precision, whose reachable fixpoints correspond to some standard direct semantics. We define and discuss two approaches to combining two given update relations to yield a third update relation, an approach based on algorithmically motivated update relations and an approach based on *merging* maximally fine-grained update relations. For both approaches, we examine how to obtain update relations for the complete labeling by combining update relations for the preferred and grounded labelings. Furthermore, we have defined novel semantics using the merge approach, including a semantics that meaningfully combines features of naive-based and complete-based semantics.

This chapter gives rise to various topics for further research. Concerning the combination of argumentation semantics, many questions remain. Further new semantics can be defined using our approach, and properties of the newly defined semantics can be studied systematically using the principle-based approach [50, 10].

Though we introduced our update relations to combine argumentation semantics, we believe that this dynamic semantics framework can be used for other applications as well. Most importantly, one of the main challenges in formal argumentation is the gap between graph based semantics and dialogue theory. Our more dynamic semantics framework may be used to decrease or even close the gap. In particular, in dialogue each statement may increase the knowledge and thus the set of arguments of participants. This is also related to the formalization of learning in the context of formal argumentation. Moreover, an important approach in argumentation semantics is the SCC recursive scheme. This scheme can be represented naturally using update relations. Various algorithms have been proposed for argumentation semantics, and these algorithmic approaches may be expressed naturally using update relations. Work has also been done on dynamic modifications to be made on a framework in order to enforce a certain set of arguments to become an extension, or prevent it from being so [54, 55]. Parallels could be made between their work and the combination operation presented in this chapter. Finally, the principle based analysis of argumentation semantics can be extended to the more fine grained update relations.

Chapter 5

An enriched argumentation framework with higher-level relations

5.1 Introduction

Dung's argumentation frameworks (AFs) [4] are a powerful and flexible formal tool for formally modelling argumentative discourse. However, various researchers have felt the need to extend AFs in order to model features of argumentation that cannot be directly modeled in AFs, e.g. by enriching them with recursive (higher-order) attacks [23], joint attacks [24], a support relation between arguments [25, 26], or explanatory features [27].

One technique that has already previously proven useful to study and combine such extensions is the meta-argumentation methodology involving the notion of a *flattening* [56]. A flattening is a function that maps some extended variant of argumentation frameworks into standard AFs. If there exists a definition of the various argumentation semantics for that extended variant of AFs that is independent from the definition of the flattening function, one wants the flattening to satisfy the property that it preserves these semantics, in the sense that applying the flattening function, then calculating the extensions according to some argumentation semantics, and finally unflattening the extensions should yield the same result as directly calculating the extensions according to the corresponding argumentation semantics for the extended variant of argumentation frameworks. However, flattenings can also be used to define argumentation semantics for extended variants of AFs for which there is no definition of the semantics independent of flattenings. This approach has proven particularly useful for combining multiple extensions of AFs [56], because in this case, it is often much clearer what the "right" definition of a flattening is than what the "right" direct definition of the various argumentation semantics is.

Previous work on flattening argumentation frameworks with recursive attacks (AFRAs) was limited to second-order attacks [56, 57], even though the original definition of recursive attacks was for arbitrarily deeply nested higher-order attacks [23]. This means that for the purpose of defining the flattening, attacking an attack between two arguments was allowed, but attacking such a second-order attack was already not allowed. In Section 5.3, we show how to define a flattening of arbitrary AFRAs, and prove that it conforms with the direct definition of the semantics of AFRAs.

The labelling approach for abstract argumentation has become quite popular [13], where a funciton assign one of three labels to each argument from the argumentation framework. Acceptable arguments are labelled *in*, any arguments they attack are labelled *out*, and remaining arguments are labelled *undec*. This methodology allows for a more local evaluation. Such an approach is missing for the explanatory argumentation frameworks [27], and therefore in this chapter we provide a labelling semantics for EAFs and prove it corresponds to the extension-based approach.

We then propose an enriched formalism which incorporates attacks, explanations, necessary and deductive supports, and incompatibility originating from and targeting sets of any kind of elements. We extend the labelling semantics for EAFs towards EEAFs.

The rest of the chapter is devoted to applying the meta-argumentation methodology of flattening and unflattening in order to provide a second approach to evaluating the enriched framework EEAF. We then show that there is a correspondence between the flattening semantics and the labelling semantics for EEAFs.

The explanatory relation from EAFs cannot be easily flattened. Therefore, for defining the semantics of EEAFs, we apply the meta-argumentation methodology by allowing the output of the flattening function to be an EAF rather than an AF. In other words, we flatten away recursive attacks, joint attacks and the support relation, but we do not flatten away explanations, instead making use of the semantics of EAFs instead of the semantics of standard AFs.

Finally, we illustrate the applicability of EEAFs by using them to model a piece of argumentation from the introduction to Hartry Field's book *Saving Truth from Paradox* [58], an important, relatively recent, monograph about semantic paradoxes, a major research topic within the field of philosophical logic.

The rest of the chapter is structured as follows: In Section 5.2, we introduce a labeling semantics for EAFs and show its correspondence to the extension-based semantics. In Section 5.3, we extend the meta-argumentation methodology to arbitrarily deeply nested AFRAs. In Section 5.4, we further extend this methodology to formally define the semantics of EEAFs. Section 5.6 presents examples that illustrate the applicability of EEAFs. After discussing related work in Section 5.5, we conclude the chapter in Section 5.7.

5.2 Explanatory Argumentation Framework Labelings

In this section, we present a labeling semantics for argumentative cores in EAFs and show its correspondence to the extension-based semantics. This labeling semantics of EAFs will be expanded to a labeling semantics of Extended Explanatory Argumentation Frameworks (EEAFs) in Section 5.4.

A labeling of an EAF is a pair consisting of two functions: One which labels arguments and another which labels the elements of the explanation relation $\dashrightarrow$. Each argument will be assigned one of the three standard labels *in*, *out* or *undec*, while each pair $(x, y) \in \dashrightarrow$ is only assigned one of two labels: `exp` when x explains y, and `nexp` when x does not explain y.

Definition 5.2.1 (EAF Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. A *labeling* of F

is a pair $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$, where $\mathit{Lab}_{\mathcal{A}}$ is a function from $\mathcal{A}$ to $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$, and $\mathit{Lab}_{\dashrightarrow}$ is a function from $\dashrightarrow$ to $\{\texttt{exp}, \texttt{nexp}\}$.

We define a notion of legal label, in the same way that the complete labeling is define via the legal of individual argument labels. In the case of EAFs however, we also have the addition of the incomparability relation, which prevents two incompatible arguments from being *in*, but is not enough on its own to justify either of them being *out*. We additionally have the labels on the pairs $(x, y) \in \dashrightarrow$, which take the value exp whenever x is labeled *in*.

Definition 5.2.2 (Legal EAF Label)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF, $b \in \mathcal{A}$, $x \in \mathcal{X}$ and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that:

- b is *legally in w.r.t. Lab* iff for every argument c attacking b, $\mathit{Lab}_{\mathcal{A}}(c) = \mathit{out}$ and for every argument d incompatible with b, $\mathit{Lab}_{\mathcal{A}}(d) \neq \mathit{in}$;
- b is *legally out w.r.t. Lab* iff there exists an argument c attacking b such that $\mathit{Lab}_{\mathcal{A}}(c) = \mathit{in}$;
- b is *legally undec w.r.t. Lab* iff it is not legally *in* nor legally *out*;
- (x, y) is *legally* $\texttt{exp}$ *w.r.t. Lab* iff $\mathit{Lab}_{\mathcal{A}}(x) = \mathit{in}$;
- (x, y) is *legally* $\texttt{nexp}$ *w.r.t. Lab* iff it is not legally $\texttt{exp}$.

We lift this notion of individual legal labels to the notions of admissible and complete labelings for the entire framework.

Definition 5.2.3 (Admissible EAF Label)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that *Lab* is a *admissible labeling of F* iff the following conditions hold:

- If $b \in \mathcal{A}$ and $\mathit{Lab}_{\mathcal{A}}(b) = \mathit{in}$, then b is legally *in* w.r.t. *Lab*.
- If $b \in \mathcal{A}$ and $\mathit{Lab}_{\mathcal{A}}(b) = \mathit{out}$, then b is legally *out* w.r.t. *Lab*.
- If $(x, y) \in \dashrightarrow$ and $\mathit{Lab}_{\dashrightarrow}((x, y)) = \texttt{exp}$, then (x, y) is legally $\texttt{exp}$ w.r.t. *Lab*.
- If $(x, y) \in \dashrightarrow$ and $\mathit{Lab}_{\dashrightarrow}((x, y)) = \texttt{nexp}$, then (x, y) is legally $\texttt{nexp}$ w.r.t. *Lab*.

We say that *Lab* is a *complete labeling of F* iff *Lab* is an admissible labeling of F and additionally satisfies the following property:

- If $b \in \mathcal{A}$ and $\mathit{Lab}_{\mathcal{A}}(b) = \mathit{undec}$, then b is legally *undec* w.r.t. *Lab*.

An *argumentative core labeling* is now defined to be a complete labeling with a maximal set of *in*-labeled arguments and a maximal set of explained explananda:

Definition 5.2.4 (AC-labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ is an *argumentative core labeling (AC-labeling)* of F iff *Lab* is a complete labeling of F and there is no complete labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathcal{A}}, \mathit{Lab}'_{\dashrightarrow})$ of F such that $\{b \in \mathcal{A} \mid \mathit{Lab}'_{\mathcal{A}}(b) = \mathit{in}\} \supsetneq \{b \in \mathcal{A} \mid \mathit{Lab}_{\mathcal{A}}(b) = \mathit{in}\}$ or $\{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}'_{\dashrightarrow}((b, e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}_{\dashrightarrow}((b, e)) = \texttt{exp}\}$.

For defining the *explanatory core labeling*, we first need the notion of *having explanatory relevance*, which captures the idea of an exp-labeled pair in the explanation relation that contributes to the explanation of an explanandum through a path of exp-labeled pairs in the explanation relation:

Definition 5.2.5 (Explanatory Relevance)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF, let $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F, and let $(x, y) \in \dashrightarrow$. We say that (x, y) *has explanatory relevance w.r.t. Lab* iff $\mathit{Lab}_{\dashrightarrow}((x, y)) = \texttt{exp}$ and there is a path $(y_0, \ldots, y_n)$ such that $y_0 = y$, $y_n \in \mathcal{X}$ and for every $0 \leq i < n$, $(y_i, y_{i+1}) \in \dashrightarrow$ and $\mathit{Lab}_{\dashrightarrow}((y_i, y_{i+1})) = \texttt{exp}$.

Definition 5.2.6 (Satisfactory Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ is a *satisfactory labeling* of F iff Lab is an admissible labeling of F and there is no admissible labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathcal{A}}, \mathit{Lab}'_{\dashrightarrow})$ of F such that $\{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}'_{\dashrightarrow}((b, e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}_{\dashrightarrow}((b, e)) = \texttt{exp}\}$.

Definition 5.2.7 (Insightful Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ is an *insightful labeling* of F iff Lab is an satisfactory labeling of F and there is no satisfactory labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathcal{A}}, \mathit{Lab}'_{\dashrightarrow})$ of F such that $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$'\} \supsetneq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$\}$.

Definition 5.2.8 (EC-Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF and $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ is an *explanatory core labeling (EC-labeling)* of F iff Lab is an insightful labeling of F and there is no insightful labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathcal{A}}, \mathit{Lab}'_{\dashrightarrow})$ of F such that $\{b \in \mathcal{A} \mid \mathit{Lab}'_{\mathcal{A}}(b) = \mathit{in}\} \subsetneq \{b \in \mathcal{A} \mid \mathit{Lab}_{\mathcal{A}}(b) = \mathit{in}\}$.

For the rest of this section, we establish that AC-labelings as defined above correspond to AC-extensions as defined in Section 2.1.2, and that EC-labelings as defined above correspond to EC-extensions as defined in Section 2.1.2. For this purpose we first need to define the standard translation between labelings and sets of arguments that allows us to formally express this correspondence:

Definition 5.2.9 (`Lab2Ext`, `Lab2Expl`, `Ext2Lab`)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF, let $\mathit{Lab} = (\mathit{Lab}_{\mathcal{A}}, \mathit{Lab}_{\dashrightarrow})$ be a labeling of F, and let $S \subseteq \mathcal{A}$ be a set of arguments. We define the function `Lab2Ext` from labelings to sets of argument as follows: $\texttt{Lab2Ext}(\mathit{Lab}) = \{b \in \mathcal{A} \mid \mathit{Lab}_{\mathcal{A}}(b) = \mathit{in}\}$. Furthermore, we define the function `Ext2Lab` from sets of arguments to labelings as follows: For every $b \in \mathcal{A}$, define

$$\texttt{Ext2Lab}(S)_{\mathcal{A}}(b) = \begin{cases} \mathit{in} & \text{if } b \in S \\ \mathit{out} & \text{if there is an argument } c \in S \text{ such that } c \rightarrow b \\ \mathit{undec} & \text{otherwise} \end{cases}$$

For every $(x, y) \in \dashrightarrow$, define

$$\texttt{Ext2Lab}(S)_{\dashrightarrow}((x, y)) = \begin{cases} \texttt{exp} & \text{if } x \in S \\ \texttt{nexp} & \text{otherwise} \end{cases}$$

We need the following lemmas for our correspondence results:

Lemma 5.2.1. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. Let Lab be a admissible labeling of F. Then $\texttt{Lab2Ext}(\textit{Lab})$ is an admissible subset of $\mathcal{A}$.*

Proof. Suppose for a contradiction that $\texttt{Lab2Ext}(\textit{Lab})$ is not conflict-free. Then there are $b, c \in \texttt{Lab2Ext}(\textit{Lab})$ such that $b \rightarrow c$ or $b \sim c$. By Definition 5.2.9, $\textit{Lab}_{\mathcal{A}}(b) = \textit{Lab}_{\mathcal{A}}(c) = \textit{in}$. But then c is not legally *in*, which contradicts the fact that *Lab* is an admissible labeling. So $\texttt{Lab2Ext}(\textit{Lab})$ is conflict-free.

For proving that $\texttt{Lab2Ext}(\textit{Lab})$ defends itself, suppose $a \in \texttt{Lab2Ext}(\textit{Lab})$ and $b \rightarrow a$. By Definition 5.2.9, $\textit{Lab}_{\mathcal{A}}(a) = \textit{in}$, so a is legally *in* w.r.t. *Lab*. Thus $\textit{Lab}_{\mathcal{A}}(b) = \textit{out}$, which in turn implies that b is legally *out* w.r.t. *Lab*. But this means that there is an argument c such that $c \rightarrow b$ and $\textit{Lab}_{\mathcal{A}}(c) = \textit{in}$, i.e. $c \in \texttt{Lab2Ext}(\textit{Lab})$, as required. □

Lemma 5.2.2. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. Let E be an admissible subset of $\mathcal{A}$. Then $\texttt{Ext2Lab}(E)$ is an admissible labeling.*

Proof. We separately check the two required properties of an admissible labeling:

- Suppose $\texttt{Ext2Lab}(E)(a) = \textit{in}$, i.e. $a \in E$. Suppose $b \sim a$. By conflict-freeness of E, $b \notin E$, so $\texttt{Ext2Lab}(E)(b) \neq \textit{in}$, as required. Furthermore, suppose $c \rightarrow a$. By admissibility of E, E defends a, i.e. some argument $d \in E$ attacks c. So $\texttt{Ext2Lab}(E)(c) = \textit{out}$, as required. Thus a is legally *in* w.r.t. $\texttt{Ext2Lab}(E)$.
- Suppose $\texttt{Ext2Lab}(E)(a) = \textit{out}$. This means that there is some $b \in E$ that attacks a. Then $\texttt{Ext2Lab}(E)(b) = \textit{in}$, so a is legally *out* w.r.t. $\texttt{Ext2Lab}(E)$.

□

Lemma 5.2.3. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. Let E be a maximal admissible subset of $\mathcal{A}$. Then $\texttt{Ext2Lab}(E)$ is a complete labeling.*

Proof. By Lemma 5.2.2, we know that $\texttt{Ext2Lab}(E)$ is an admissible labeling. So in order to prove that $\texttt{Ext2Lab}(E)$ is a complete labeling, we only need to prove that any argument labeled *undec* by $\texttt{Ext2Lab}(E)$ is legally *undec* w.r.t. $\texttt{Ext2Lab}(E)$.

Suppose $\texttt{Ext2Lab}(E)(a) = \textit{undec}$. Then no argument in E attacks a, i.e. no argument b with $\texttt{Ext2Lab}(E)(b) = \textit{in}$ attacks a, i.e. a is not not legally *out* w.r.t. $\texttt{Ext2Lab}(E)$. Suppose for a contradiction that a is legally *in* w.r.t. $\texttt{Ext2Lab}(E)$. Then one can easily see that $E \cup \{a\}$ is admissibile, contradicting the maximiality of E. So a is not not legally *in* w.r.t. $\texttt{Ext2Lab}(E)$. Hence a is legally *undec* w.r.t. $\texttt{Ext2Lab}(E)$. □

Lemma 5.2.4. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If Lab is a satisfactory labeling of F, then $\texttt{Lab2Ext}(\textit{Lab})$ is a satisfactory subset of $\mathcal{A}$. Furthermore, if S is a satisfactory subset of $\mathcal{A}$, then $\texttt{Ext2Lab}(S)$ is a satisfactory labeling of F.*

Proof. This directly follows from Lemmas 5.2.1 and 5.2.2 as well as Definitions 2.1.14 and 5.2.6. □

Lemma 5.2.5. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If Lab is an insightful labeling of F, then $\texttt{Lab2Ext}(\mathit{Lab})$ is an insightful subset of $\mathcal{A}$. Furthermore, if S is an insightful subset of $\mathcal{A}$, then $\texttt{Ext2Lab}(S)$ is an insightful labeling of F.*

Proof. Suppose *Lab* is an insightful labeling of F. Then *Lab* is a satisfactory labeling, so by Lemma 5.2.4, $\texttt{Lab2Ext}(\mathit{Lab})$ is a satisfactory subset of $\mathcal{A}$.

Now suppose for a contradiction that there is a satisfactory set $S' \subseteq \mathcal{A}$ such that $S' >_d \texttt{Lab2Ext}(\mathit{Lab})$.

First, we show that $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$\} \subseteq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S')\}$. Suppose $(x, y) \in \dashrightarrow$ and (x, y) has explanatory relevance w.r.t. *Lab*. This means that there is a path $(y_0, \ldots, y_n)$ such that $y_0 = y$, $y_n \in \mathcal{X}$ and for every $0 \leq i < n$, $(y_i, y_{i+1}) \in \dashrightarrow$ and $\mathit{Lab}_{\dashrightarrow}((y_i, y_{i+1})) = \texttt{exp}$. Choose such a path of minimal length. Then $X := \{x, y, y_1, \ldots, y_{n-1}\}$ is an explanation for y_n offered by $\texttt{Lab2Ext}(\mathit{Lab})$. Since $S' >_d \texttt{Lab2Ext}(\mathit{Lab})$, there is an explanation $X' \supseteq X$ offered by S'. Then for every $0 \leq i < n$, $y_i \in S'$. So for every $0 \leq i < n$, $\texttt{Ext2Lab}(S')_{\dashrightarrow}((y_i, y_{i+1})) = \texttt{exp}$. So $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$\} \subseteq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S')\}$, as required.

Now we show that $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$\} \subsetneq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S')\}$ by showing that there is an $(x, y) \in \dashrightarrow$ that has explanatory relevance w.r.t. $\texttt{Ext2Lab}(S')$ but not w.r.t. *Lab*. To show this, note that the fact that $S' >_d \texttt{Lab2Ext}(\mathit{Lab})$ implies that there is an explanation X offered by S' such that X is not an explanation offered by $\texttt{Lab2Ext}(\mathit{Lab})$. This means that for some $a' \in X$ such that there is no $\dashrightarrow$-path from a' to an explanandum that contains only arguments from $\texttt{Lab2Ext}(\mathit{Lab})$. This in turn means that there is there is no path $(y_0, \ldots, y_n)$ such that $y_0 = a'$, $y_n \in \mathcal{X}$ and for every $0 \leq i < n$, $(y_i, y_{i+1}) \in \dashrightarrow$ and $\mathit{Lab}_{\dashrightarrow}((y_i, y_{i+1})) = \texttt{exp}$. In other words, this means that a' does not have explanatory relevance w.r.t. *Lab*. Similarly, the fact that $a' \in X$ and that X is an explanation offered by S' means that a' does not have explanatory relevance w.r.t. $\texttt{Ext2Lab}(S')$.

This concludes the proof that $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$\} \subsetneq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S')\}$. This however means that *Lab* is not an insightful labeling of F, contrary to our assumption. Thus there is no satisfactory set $S' \subseteq \mathcal{A}$ such that $S' >_d \texttt{Lab2Ext}(\mathit{Lab})$, i.e. $\texttt{Lab2Ext}(\mathit{Lab})$ is an insightful subset of $\mathcal{A}$.

For the second claim of the lemma, suppose that S is an insightful subset of $\mathcal{A}$. Then S is a satisfactory subset of $\mathcal{A}$, so by Lemma 5.2.4, $\texttt{Ext2Lab}(S)$ is a satisfactory labeling of F.

Now suppose for a contradiction that there is a satisfactory labeling *Lab*$'$ of F such that $\{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t. Lab*$'\} \supsetneq \{(x, y) \in \dashrightarrow \mid (x, y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S)\}$.

We show that $\texttt{Lab2Ext}(\mathit{Lab}') >_d S$. For showing this, assume X is an explanation offered by S. Then there is a unique $a \in X$ that explains an exlpanandum e, and for every $a' \in X \setminus \{a\}$, there is a $\dashrightarrow$-path from a' to a consisting of arguments in S. Fix some $a' \in X$. If $a' = a$, then clearly (a, e) has explanatory relevance w.r.t. $\texttt{Ext2Lab}(S)$, so by the above assumption, it also has explanatory relevance w.r.t. *Lab*$'$, i.e. $a \in \texttt{Lab2Ext}(\mathit{Lab}')$.

If $a' \neq a$, then there is a $\dashrightarrow$-path $(y_0, \ldots, y_n)$ from $y_0 = a'$ to $y_n = a$ consisting of arguments in S. In this case, for every $0 \leq i < n$, (y_i, y_{i+1}) has explanatory relevance w.r.t. $\texttt{Ext2Lab}(S)$, so by the above assumption, it also has explanatory relevance w.r.t. Lab', i.e. $y_i \in \texttt{Lab2Ext}(\mathit{Lab}')$. These two facts taken together mean that $X \subseteq \texttt{Lab2Ext}(\mathit{Lab}')$, i.e. X is an explanation offered by S.

We still need to show that there is an explanation X' offered by $\texttt{Lab2Ext}(\mathit{Lab}')$ such that there is no explanation $X \supseteq X'$ offered by S. To show this, observe that our assumption that $\{(x,y) \in \dashrightarrow \mid (x,y)$ *has explanatory relevance w.r.t.* $\mathit{Lab}'\} \supsetneq \{(x,y) \in \dashrightarrow \mid (x,y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S)\}$ implies that there is some $(x,y) \in \dashrightarrow$ such that (x,y) has explanatory relevance w.r.t. Lab', but not w.r.t. $\texttt{Ext2Lab}(S)$. This means that there is a path $(y_0, \ldots, y_n)$ such that $y_0 = y$, $y_n \in \mathcal{X}$ and for every $0 \leq i < n$, $(y_i, y_{i+1}) \in \dashrightarrow$ and $\mathit{Lab}'_{\dashrightarrow}((y_i, y_{i+1})) = \texttt{exp}$. Choose such a path of minimal length. Then $X' := \{x, y, y_1, \ldots, y_{n-1}\}$ is an explanation for y_n offered by $\texttt{Lab2Ext}(\mathit{Lab}')$. But since (x,y) does not have explanatory relevance w.r.t. $\texttt{Ext2Lab}(S)$, there is no $\dashrightarrow$-path from x to an explanandum containing only arguments in S, which means that some element of X' is not in S, so there is no explanation $X \supseteq X'$ offered by S.

This concludes the proof that $\texttt{Lab2Ext}(\mathit{Lab}') >_d S$. However, this fact contradicts the assumption that S is an insightful subset of $\mathcal{A}$. Thus our assumption that there is a satisfactory labeling Lab' of F such that $\{(x,y) \in \dashrightarrow \mid (x,y)$ *has explanatory relevance w.r.t.* $\mathit{Lab}'\} \supsetneq \{(x,y) \in \dashrightarrow \mid (x,y)$ *has explanatory relevance w.r.t.* $\texttt{Ext2Lab}(S)\}$ must be false. Hence $\texttt{Ext2Lab}(S)$ is an insightful labeling of F, as required. □

Now we are ready to state and prove the correspondence results. We start with the correspondence between AC-labelings and AC-extensions:

Proposition 5.2.6. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If Lab is an AC-labeling of F, then $\texttt{Lab2Ext}(\mathit{Lab})$ is an AC-extension of F. Furthermore, if E is an AC-extension of F, then $\texttt{Ext2Lab}(E)$ is an AC-labeling of F.*

Proof. Suppose *Lab* is an AC-labeling of F. In this case *Lab* is a complete labeling, and therefore also an admissible labeling of F. So by Lemma 5.2.1, $\texttt{Lab2Ext}(\mathit{Lab})$ is an admissible subset of $\mathcal{A}$.

For showing that $\texttt{Lab2Ext}(\mathit{Lab})$ is satisfactory, we assume for a contradiction that there is a set $S \subseteq \mathcal{A}$ such that $S >_p \texttt{Lab2Ext}(\mathit{Lab})$ and S is admissible. Choose a maximal set $S' \supseteq S$ that is still admissible. By Lemma 5.2.3, $\texttt{Ext2Lab}(S')$ is a complete labeling of F. Since $S >_p \texttt{Lab2Ext}(\mathit{Lab})$ and $S' \supseteq S$, we have $S' >_p \texttt{Lab2Ext}(\mathit{Lab})$. But then $\{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \texttt{Ext2Lab}(S')_{\dashrightarrow}((b,e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}_{\dashrightarrow}((b,e)) = \texttt{exp}\}$, contradicting the assumption that *Lab* is an AC-labeling of F. Thus $\texttt{Lab2Ext}(\mathit{Lab})$ is satisfactory.

Assume for a contradiction that there is some $S \supsetneq \texttt{Lab2Ext}(\mathit{Lab})$ such that S is satisfactory. Then S is admissible. Choose a maximal set $S' \supseteq S$ that is still admissible. By Lemma 5.2.3, $\texttt{Ext2Lab}(S')$ is a complete labeling. Since $S' \supseteq S \supsetneq \texttt{Lab2Ext}(\mathit{Lab})$, this contradict the assumption that *Lab* is an AC-labeling of F. Thus $\texttt{Lab2Ext}(\mathit{Lab})$ is an AC-extension of F.

For the second statement of this proposition, suppose that E is an AC-extension of F. Since E is admissible, $\texttt{Ext2Lab}(E)$ is a complete labeling by Lemma 5.2.3. Suppose for a contradiction that there is a complete labeling *Lab*$'$ such that $\{b \in \mathcal{A} \mid \mathit{Lab}'_{\mathcal{A}}(b) = \mathit{in}\} \supsetneq \{b \in \mathcal{A} \mid \texttt{Ext2Lab}(E)_{\mathcal{A}}(b) = \mathit{in}\}$ or $\{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}'_{\dashrightarrow}((b,e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \texttt{Ext2Lab}(E)_{\dashrightarrow}((b,e)) = \texttt{exp}\}$. Since *Lab*$'$ is an admissible labeling, $\texttt{Lab2Ext}(\mathit{Lab}')$ is an admissible subset of $\mathcal{A}$ by Lemma 5.2.1. We now consider the two cases separately:

1. $\{b \in \mathcal{A} \mid \mathit{Lab}'_{\mathcal{A}}(b) = \mathit{in}\} \supsetneq \{b \in \mathcal{A} \mid \texttt{Ext2Lab}(E)_{\mathcal{A}}(b) = \mathit{in}\}$. This means that $\texttt{Lab2Ext}(\mathit{Lab}') \supsetneq E$. Since E is satisfactory, this implies that $\texttt{Lab2Ext}(\mathit{Lab}')$ is satisfactory too. But this in turn means that E is not maximal among the satisfactory sets, contradicting the assumption that E is an AC-extension of F.

2. $\{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \mathit{Lab}'_{\dashrightarrow}((b,e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathcal{A}, \texttt{Ext2Lab}(E)_{\dashrightarrow}((b,e)) = \texttt{exp}\}$. This means that $\texttt{Lab2Ext}(\mathit{Lab}') >_p E$, contradicting the fact that E is satisfactory.

Thus there is no such complete labeling *Lab*$'$, so $\texttt{Ext2Lab}(E)$ is indeed an AC-labeling of F. □

Now we consider the correspondence between EC-labelings and EC-extensions:

Proposition 5.2.7. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If Lab is an EC-labeling of F, then $\texttt{Lab2Ext}(\mathit{Lab})$ is an EC-extension of F. Furthermore, if E is an EC-extension of F, then $\texttt{Ext2Lab}(E)$ is an EC-labeling of F.*

Proof. This directly follows from Lemma 5.2.5 as well as Definitions 2.1.14 and 5.2.7. □

Concerning the relation between EC-labelings and AC-labelings, observe that both require admissibility and maximize explanatory power. The difference lies in that AC-labelings maximize *in* arguments, while EC-labelings first maximize explanatory depth, but then minimize *in* arguments. So it turns out that every EC-labeling can be turned into an AC-labeling by changing the label of some *undec* arguments to *in* or *out*.

Proposition 5.2.8. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If E is an EC-extension of F, then there exists an AC-extension E' such that $E \subseteq E'$.*

Proof. Let E be an EC-extension of F. By Definition 2.1.14, E is satisfactory. Now we distinguish two cases:

1. E is an AC-extension. Then, we are done, since $E \subseteq E$.

2. E is not an AC-extension. Then, suppose for a contradiction that there is no AC-extension E' such that $E \subseteq E'$. Then, by Definition 2.1.14, E is an AC-extension. This is a contradiction, therefore there exists an AC-extension E' such that $E \subseteq E'$.

So in both cases we have that there exists an AC-extension E' such that $E \subseteq E'$. □

Corollary 5.2.9. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ be an EAF. If Lab is an EC-labeling of F, then there exists an AC-labeling Lab$'$ such that for all $a \in \mathcal{A}$, if $\mathit{Lab}(a) \in \{\mathit{in}, \mathit{out}\}$, then $\mathit{Lab}'(a) = \mathit{Lab}(a)$.*

Proof. The *in* part of the corollary follows from Propositions 5.2.6, 5.2.7 and 5.2.8, while the *out* part follows from both AC-labelings and EC-labeling being admissible. □

5.3 Flattening AFRAs

In order to motivate the semantics of the framework we will present in section 5.4 based on a flattening function, we will start by suggesting a flattening for AFRAs of any order. We will prove that this flattening leads to the same extensions as the AFRA semantics defined by Baroni et al. [23].

Boella et al. [56] define a flattening function for second-order AFRAs, which allows one to obtain for a given AFRA an equivalent abstract argumentation framework. We will now propose a flattening function for AFRAs of any order.

We will first define a function m which will associate each argument and each attack relation to the corresponding meta-argument. For an argument a, it will be the meta-argument $acc(a)$, while for an attack, it will be the Y auxiliary argument, since its acceptability is synonym of success for the attack.

Definition 5.3.1 (AFRA Meta-Arguments)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA. The set of corresponding meta-arguments is $MA = \{acc(a) \mid a \in \mathcal{A}\} \cup \{X_{a,\psi}, Y_{a,\psi} \mid a \in \mathcal{A},\ \psi \in (\mathcal{A} \cup \rightarrow)\}$. We define a partial function m: $(\mathcal{A} \cup \rightarrow) \mapsto MA$, such that:

- if $\varphi \in \mathcal{A}$, then $m(\varphi) = acc(\varphi)$.
- if $\varphi \in \rightarrow$ such that for some $\psi \in \mathcal{A}$ and some $\delta \in (\mathcal{A} \cup \rightarrow)$, $\varphi = (\psi, \delta)$, then $m(\varphi) = Y_{\psi,\delta}$.

We define the *flattening function* f to be $f(F) = \langle MA, \rightarrow_2 \rangle$, where $\rightarrow_2 \subseteq MA \times MA$ is a binary relations on MA such that

$$acc(a) \rightarrow_2 X_{a,\psi}, X_{a,\psi} \rightarrow_2 Y_{a,\psi} \text{ and } Y_{a,\psi} \rightarrow_2 m(\psi) \text{ for all } a \in \mathcal{A}, \psi \in (\mathcal{A} \cup \rightarrow)$$

One can then apply the classical abstract argumentation semantics such as complete, stable, preferred and grounded. We then need to define a function which can transform a meta-extension from the flattened AFRA to an extension for the original AFRA. A similar unflattening function has been introduced in [56], and has been slightly modified here to also unflatten attacks.

Definition 5.3.2 (AFRA Unflattening Function)**.** Given a set of meta-arguments $B \subseteq MA$, we define the *unflattening function* g as:

$$g(B) = \{a \mid acc(a) \in B\} \cup \{(a, \psi) \mid Y_{a,\psi} \in B\}$$

We also define a function $\bar{f}$ which provides a correspondence between a set of arguments and attacks from an AFRA and a set of meta-arguments from its flattened version.

Definition 5.3.3 (AFRA Correspondence Function)**.** Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ its flattening and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We define the *correspondence function* $\bar{f}: \mathbb{P}(\mathcal{A} \cup \rightarrow) \mapsto \mathbb{P}(MA)$ as follows:

$$\bar{f}(S) = \{acc(a) \mid a \in S \cap \mathcal{A}\} \cup \{Y_{a,\psi} \mid (a, \psi) \in S \cap \rightarrow\} \cup \\ \{X_{b,\psi} \mid (a, b) \in S \cap \rightarrow,\ \psi \in \rightarrow\}$$

Notice that $g(\bar{f}(S)) = S$. We add the extra $X_{i,j}$ meta-arguments in order to represent the indirect attacks which the arguments in S might carry out, i.e. the attacks which are indirectly attacked by arguments in S due to them attacking the source of these attacks.

In [23], Baroni et al. define the semantics of AFRAs without having recourse to flattening. We will show that the process of flattening, applying complete semantics on the flattened frameworks and then unflattening it is equivalent to the directly applying the semantics they define for the complete semantics. We will show this gradually by first stating and proving three lemmas:

Lemma 5.3.1. *Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. S is conflict-free in F if and only if $\bar{f}(S)$ is conflict-free in $f(F)$.*

Proof. Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$.

1. $\Rightarrow$: Assume that S is conflict-free in F. Then, there is no $\varphi, \psi \in S$ such that $trg(\varphi) = \psi$ or $trg(\varphi) = src(\psi)$. Suppose for a contradiction that $\bar{f}(S)$ is not conflict-free in $f(F)$. This means that there exists two arguments $p, q \in \bar{f}(S)$ such that $p \rightarrow_2 q$. By the construction of $\rightarrow_2$ defined by the flattening function, there are only four possible cases, which all lead to the contradiction that S is not conflict-free. Therefore $\bar{f}(S)$ is conflict-free.

2. $\Leftarrow$: Suppose $\bar{f}(S)$ is conflict-free. Suppose for a contradiction that S is not conflict-free. Then, there exists $(a, \varphi), (b, \psi) \in S$ such that $\varphi = (b, \psi)$ or $\varphi = b$.

   In both cases we can reach the contradiction that $\bar{f}(S)$ is not conflict-free, therefore S is conflict-free.

□

Lemma 5.3.2. *Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$, $\varphi \in (\mathcal{A} \cup \rightarrow)$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We have that:*

φ is defended by S in F and if $\varphi = (a, \psi) \in \rightarrow$, we have $a \in S$, iff $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.

Proof. Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$, $\varphi \in (\mathcal{A} \cup \rightarrow)$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$.

1. $\Rightarrow$: Suppose that φ is defended by S in F and if $\varphi = (a, \psi) \in \rightarrow$, we have $a \in S$. Consider $m(\varphi)$ in $f(F)$. Suppose for some $p \in MA$, $p \rightarrow_2 m(\varphi)$. By the construction of $\rightarrow_2$ defined by the flattening function, either $p = Y_{a,\varphi}$ for some $a \in \mathcal{A}$, or $p = X_{src(\varphi),trg(\varphi)}$.

   In both cases, $m(\varphi)$ is defended by $\bar{f}(S)$. Hence, if φ is defended by S in F, then $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$. We now have to show that if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.
   Suppose $\varphi \in \rightarrow$ and $p \in MA$ such that $p \rightarrow_2 acc(src(\varphi))$. Then, p must be of

the form $Y_{a,src(\varphi)}$ for some $a \in \mathcal{A}$, and hence there exists $(a, src(\varphi)) \in \rightarrow$. Since $(a, src(\varphi))$ defeats φ, there exists some $\delta \in S$ such that δ defeats $(a, src(\varphi))$. We distinguish two cases:

Either $\delta = (b, a)$ or $\delta = (b, (a, src(\varphi)))$ for some $b \in \mathcal{A}$. In both cases, $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.
Therefore, if φ is defended by S in F and if $\varphi = (a, \psi) \in \rightarrow$, we have $a \in S$, then $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.

2. $\Leftarrow$: Suppose $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$. Consider φ in F. Suppose that for some $\psi \in \rightarrow$, ψ defeats φ. This means that either $\psi = (a, \varphi)$ or $\psi = (a, src(\varphi))$ for some $a \in \mathcal{A}$. In both cases, we can conclude that there exists a $\delta \in S$ such that δ defeats ψ by contradiction. Therefore, φ is defended by S.

   We now have to show that if $\varphi = (a, \psi) \in \rightarrow$, we have $a \in S$, still under the assumption that $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.
   Suppose that $\varphi = (a, \psi) \in \rightarrow$. Then, by the construction of $\rightarrow_2$ defined by the flattening function, we have $X_{a,\psi} \rightarrow_2 Y_{a,\psi}$. Since $m(\varphi) = Y_{a,\psi}$ is defended by $\bar{f}(S)$, there exists $p \in \bar{f}(S)$ such that $p \rightarrow_2 X_{a,\psi}$. By the construction of $\rightarrow_2$, the only possibility is $p = acc(a)$. Hence, $acc(a) \in \bar{f}(S)$. Therefore, we have $a \in S$.

Thus, we can conclude that φ is defended by S in F and if $\varphi = (a, \psi) \in \rightarrow$, we have $a \in S$, if and only if $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$.

□

Lemma 5.3.3. *Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. We have that:*

S is admissible in F and for every $(a, \psi) \in (S \cap \rightarrow)$, we have that $a \in S$
if and only if
$\bar{f}(S)$ is admissible in $f(F)$.

Proof. Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$.

1. $\Rightarrow$:Suppose $\bar{f}(S)$ is admissible in $f(F)$. Then, $\bar{f}(S)$ is conflict-free. Hence, according to Lemma 5.3.1, S is also conflict-free.
   Let $\varphi \in S$. We need to show that φ is defended by S. We do this by applying Lemma 5.3.2, i.e. by establishing that $m(\varphi)$ is defended by $\bar{f}(S)$ in $f(F)$ and if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is also defended by $\bar{f}(S)$. We have $m(\varphi) \in \bar{f}(S)$ and $m(\varphi)$ is defended by $\bar{f}(S)$ since $\bar{f}(S)$ is admissible. By the definition of $\bar{f}$, for every $(a, \psi) \in (S \cap \rightarrow)$, we have $Y_{a,\psi} \in \bar{f}(S)$. Therefore, $acc(a) \in \bar{f}(S)$, since it is the only argument which can defend $Y_{a,\psi}$ from $X_{a,\psi}$'s attack and $\bar{f}(S)$ is admissible. This means that $acc(a)$ is defended by $\bar{f}(S)$. Thus, according to Lemma 5.3.2, every $\varphi \in S$ is defended by

S, which means that S is admissible, and for every $(a, \psi) \in (S \cap \rightarrow)$, we have that $a \in S$.

2. $\Leftarrow$:Suppose S is admissible in F and for every $(a, \psi) \in (S \cap \rightarrow)$, we have that $a \in S$. Then, S is conflict-free and so, according to Lemma 5.3.1, $\bar{f}(S)$ is also conflict-free.
   Let $p \in \bar{f}(S)$. p is either of the form $m(\varphi)$ for some $\varphi \in S$, or of the form $X_{a,b}$ for some $a, b \in MA$ and $(\psi, a) \in S$.

   In both cases, p is defended by $\bar{f}(S)$. Hence, $\bar{f}(S)$ is admissible in $f(F)$.

Therefore, S is admissible in F and for every $(a, \psi) \in (S \cap \rightarrow)$, we have that $a \in S$, if and only if $\bar{f}(S)$ is admissible in $f(F)$. □

Theorem 5.3.4. *Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$. S is a complete extension of F if and only if $\bar{f}(S)$ is a complete extension of $f(F)$.*

Proof. Let $F = \langle \mathcal{A}, \rightarrow \rangle$ be an AFRA, $f(F) = \langle MA, \rightarrow_2 \rangle$ and $S \subseteq (\mathcal{A} \cup \rightarrow)$.

1. $\Rightarrow$: Suppose S is a complete extension of F. For every $(a, \psi) \in (S \cap \rightarrow)$, by the definition of defeat, a is defended by S, and thus $a \in S$. Therefore, by Lemma 5.3.3, $\bar{f}(S)$ is admissible.

   Take some arbitrary $p \in MA$ and suppose that p is defended by $\bar{f}(S)$. Then, either $p = m(\varphi)$ for some $\varphi \in (\mathcal{A} \cup \rightarrow)$, or $p = X_{a,b}$ for some $a, b \in \mathcal{A}$.

   (a) Suppose that $p = m(\varphi)$ for some $\varphi \in (\mathcal{A} \cup \rightarrow)$. Now assume that $\varphi \in \rightarrow$. Then, $m(\varphi) = Y_{src(\varphi),trg(\varphi)}$. By construction of $\rightarrow_2$, we have that $X_{src(\varphi),trg(\varphi)} \rightarrow_2 Y_{src(\varphi),trg(\varphi)}$. The only argument which can defend $Y_{src(\varphi),trg(\varphi)}$ from $X_{src(\varphi),trg(\varphi)}$ is $acc(src(\varphi))$. Since $\bar{f}(S)$ defends $Y_{src(\varphi),trg(\varphi)}$, we have that $acc(src(\varphi)) \in \bar{f}(S)$. As $\bar{f}(S)$ is admissible, $acc(src(\varphi))$ is defended by $\bar{f}(S)$. Hence, if $\varphi \in \rightarrow$, then $acc(src(\varphi))$ is defended by $\bar{f}(S)$.
   Therefore, by Lemma 5.3.2, φ is defended by S. Since S is a complete extension, this means that $\varphi \in S$. Therefore, $p = m(\varphi) \in \bar{f}(S)$.

   (b) Now suppose that $p = X_{a,b}$ for some $a, b \in \mathcal{A}$. According to our assumptions, $\bar{f}(S)$ defends $X_{a,b}$. By construction of $\rightarrow_2$, the only argument which attacks $X_{a,b}$ is $acc(a)$. Hence, there exists $Y_{c,a} \in \bar{f}(S)$ for some $c \in \mathcal{A}$. So, by definition of $\bar{f}$, we have that $p = X_{a,b} \in \bar{f}(S)$.

   In either case, we have that $p \in \bar{f}(S)$. Hence, $\bar{f}(S)$ contains all arguments it defends. Since it is also admissible, $\bar{f}(S)$ is a complete extension of $f(F)$.

2. $\Leftarrow$:Suppose that $\bar{f}(S)$ is a complete extension of $f(F)$. Then, $\bar{f}(S)$ is admissible and contains all arguments it defends. According to Lemma 5.3.3, we have that S is admissible and for every $(a, \psi) \in (S \cap \rightarrow)$, we have that $a \in S$. Suppose that for some $\varphi \in (\mathcal{A} \cup \rightarrow)$, φ is defended by S. Hence, by Lemma 5.3.2, $m(\varphi)$ is defended by $\bar{f}(S)$. Since $\bar{f}(S)$ is a complete extension of $f(F)$, $m(\varphi) \in \bar{f}(S)$. Hence, by construction of $\bar{f}(S)$, we have that $\varphi \in S$. Therefore, for any $\varphi \in (\mathcal{A} \cup \rightarrow)$ such that φ is defended by S, we have $\varphi \in S$. Since S is also admissible, S is a complete extension of F.

Hence, S is a complete extension of F if and only if $\bar{f}(S)$ is a complete extension of $f(F)$.
□

5.4 Aggregating multiple extensions of abstract argumentation frameworks: EEAFs

In this section, we will introduce Extended Explanatory Argumentation Frameworks (EEAFs), an extension of EAFs from Section 2.1.2 with meta-argumentation features such as higher order attacks, support, joint attacks and allowing these relations to originate and target arbitrary sets of elements.

We first provide a definition of EEAFs, before diving into the flattening semantics of each relation individually in the coming sub-sections. in the definition of EEAFs, we make use of four designated constants, *att*, *expl*, *dsup* and *nsup*, that serve as labels for the attack relation, the explanation relation, the deductive support relation and the necessary support relation respectively. These serve as markers to differentiate two relations between the same elements. For example, we could have an argument explaining another and simultaneously providing necessary support for it, while a third argument attacks the explanation but not the necessary support between them. Such a scenario would not be representable without the markers, as both the explanation and the necessary support would end up being the same mathematical object.

Definition 5.4.1 (Extended Explanatory Argumentation Framework (EEAF))**.** An *extended explanatory argumentation framework* (EEAF) is a tuple $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$, where $\mathcal{A}$ is a set of arguments, $\mathcal{X}$ is a set of explananda, $\rightarrow \subseteq \mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \times (\mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \setminus \emptyset) \times \{\mathit{att}\}$ is a higher-order set attack relation, $\dashrightarrow \subseteq \mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \times (\mathbb{P}(\mathcal{A} \cup \mathcal{X} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \setminus \emptyset) \times \{\mathit{expl}\}$ is a higher-order set explanatory relation, $\sim \subseteq \mathbb{P}(\mathcal{A} \cup \mathcal{X} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \setminus \emptyset$ is an incompatibility relation, $\Rightarrow_d \subseteq (\mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \setminus \emptyset) \times \mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \times \{\mathit{dsup}\}$ is a higher-order set deductive support relation and $\Rightarrow_n \subseteq \mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \times (\mathbb{P}(\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n) \setminus \emptyset) \times \{\mathit{nsup}\}$ is a higher-order set necessary support relation.

EEAFs allow for the representation of relations of different types within the same framework, providing more expressivity to the representation. In addition, EEAFs also allow for these relations to originate and target sets of other elements. For example, one could have an argument attacking a set of two other arguments, while another two arguments jointly support this attack in a deductive manner. The aim is to have a framework which is as general as possible, and therefore the framework features a minimal amount of restrictions on the interactions between the different elements.

We first presented the framework itself, but the semantics will be described only later as we first focus on the behavior of each relation. These individual behaviors are later aggregated for the definition of the EEAF semantics.

In some cases, we exclude the empty set from either the potential sources or targets of a relation. In the case of the attack relation, if we did allow for the empty set as a target, this would mean that in case of success of this attack, the empty set could not be included in any extension since it is not defended. But then this would mean that there could not be any extensions, which is a consequence that cannot be expressed at the level of an abstract argumentation graph itself, only at the level of the semantics. The flattening approach is thus unable to capture this interaction, but we argue that this phenomenon is also unintuitive, as the empty set should always be admissible, and so we do not wish for our framework to allow it. Similarly, in the case of incompatibility, having the empty set be incompatible would mean that the empty set is not conflict-free, and therefore no set of arguments would be conflict-free, leading once again to the absence of any extension. In the cases of the supports, we exclude the empty set from the target for deductive support and from the source for necessary support, since the support can be intermediarily represented as higher-order set attacks (inverted in the case of deductive support), and thus should follow the same restrictions. Regarding the explanation relation, the case is a bit different since it is a purely positive relation. Following our interpretation of the other relations, explaining the empty set would mean that there is an explanation for the environment, that any set of arguments would be explanatory deeper by including this explanation. This is again an interaction which cannot be captured inside the framework itself but only at the level of the semantics, and therefore we also exclude this possibility in our framework.

5.4.1 Labelling semantics of EEAFs

Definition 5.4.2 (Source and Target)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF. Let $\varphi = (S_1, S_2, l) \in \rightarrow \cup \dashrightarrow \cup \Rightarrow_d \cup \Rightarrow_n$. Then we call S_1 the *source of* φ, denoted $\mathsf{src}(\varphi)$, and we call S_2 the *target of* φ, denoted $\mathsf{trg}(\varphi)$.

Definition 5.4.3 (Elements and Potential Explanation Steps)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF. Any $x \in \mathcal{A} \cup \mathcal{X} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n$ is called an *element* of F. The set of elements of F is denoted as *Elms*(F). The set $\mathcal{A} \cup \rightarrow \cup \dashrightarrow \cup \sim \cup \Rightarrow_d \cup \Rightarrow_n$ is denoted *NonEx*(F). The *set of potential explanation steps of* F, denoted *PES*(F), is defined to be *PES*$(F) := \{(\varphi, x) \in \dashrightarrow \times \mathit{Elms}(F) \mid x \in \mathsf{trg}(\varphi)\}$.

Definition 5.4.4 (Preconditions)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF. Let $x \in \mathit{NonEx}(F)$. The set of preconditions of x, denoted $\mathsf{pre}(x)$, is defined as follows:

$$\mathsf{pre}(x) := \begin{cases} \emptyset & \text{if } x \notin \rightarrow \cup \dashrightarrow \\ \mathsf{src}(x) & \text{if } x \in \rightarrow \cup \dashrightarrow \end{cases}$$

Definition 5.4.5 (EEAF Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF. A *labeling* of F is a pair $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$, where $\mathit{Lab}_{\mathit{NonEx}}$ is a function from *NonEx*(F) to $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$ and $\mathit{Lab}_{\mathit{PES}}$ is a function from *PES*(F) to $\{\mathtt{exp}, \mathtt{nexp}\}$. Given a labeling $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ of F, an element x of F and a potential explanation step $(\varphi, y) \in \mathit{PES}(F)$, $\mathit{Lab}_{\mathit{NonEx}}(x)$ is called the *acceptance label* of x w.r.t. *Lab*, and $\mathit{Lab}_{\mathit{PES}}((\varphi, y))$ is called the *explanatory label* of (φ, y) w.r.t. *Lab*.

Intuitively, for a potential explanation step (φ, y) to have the explanatory label $\mathtt{exp}$ means that the source of φ actually explains y rather that one of the other elements of the target of φ, whereas for (φ, y) to have the explanatory label `nexp` means that the source of φ does not actually explain y, because it explain another element of the target of φ.

We proceed by defining a notion of legally *in* and legally *out* for each relation separately, so here we introduce the notion of being legally labelled with respect to a particular relation element. We then combine these into a global notion of legal label, resolving label conflicts with respect to different relation elements by having *out* take priority over *in*. We finally define the notion of legally *undec* as neither legally *in* nor legally *out*.

The attack relation

Definition 5.4.6 (Legal label w.r.t. one attack)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F. Let $b \in NonEx(F)$ be such that there is an attack $\varphi \in\rightarrow$ with $b \in \mathsf{trg}(\varphi)$. We say that

- b is *legally in w.r.t. Lab and attack* φ iff some element of $\{\varphi\} \cup \mathsf{src}(\varphi) \cup (\mathsf{trg}(\varphi) \setminus \{b\})$ has the acceptance label *out* w.r.t. *Lab*.
- b is *legally out w.r.t. Lab and attack* φ iff every element of $\{\varphi\} \cup \mathsf{src}(\varphi) \cup (\mathsf{trg}(\varphi) \setminus \{b\})$ has the acceptance label *in* w.r.t. *Lab*.

Definition 5.4.7 (Legal label w.r.t. attacks)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F. We say that

- b is *legally in w.r.t. Lab and attacks* iff for every attack $\varphi \in\rightarrow$ with $b \in \mathsf{trg}(\varphi)$, b is legally *in* w.r.t. *Lab* and φ.
- b is *legally out w.r.t. Lab and attacks* iff for some attack $\varphi \in\rightarrow$ with $b \in \mathsf{trg}(\varphi)$, b is legally *out* w.r.t. *Lab* and φ.

The explanation relation

Definition 5.4.8 (Legal explanation label)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let $(\varphi, b) \in PES(F)$. We say that

- (φ, b) is *legally* `exp` *w.r.t. Lab* iff every element of $\mathsf{src}(\varphi) \cup \{\varphi\}$ has the acceptance label *in* w.r.t. *Lab* and for every $x \in \mathsf{trg}(\varphi) \setminus \{b\}$, (φ, x) has the explanatory label `nexp`.
- (φ, b) is *legally* `nexp` *w.r.t. Lab* iff it is not legally `exp` w.r.t. *Lab* and φ.

The incompatibility relation

Definition 5.4.9 (Legal label w.r.t. one incompatibility)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F such that there is a set $\varphi \in\sim$ of incompatible elements with $b \in \varphi$. We say that b is *legally in w.r.t. Lab and* φ iff some element of $\varphi \setminus \{b\}$ does not have the acceptance label *in* w.r.t. *Lab*.

Definition 5.4.10 (Legal label w.r.t. incompatibilities)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F. We say that b is *legally in w.r.t. Lab and incompatibilites* iff for every set $\varphi \in\sim$ with $b \in \varphi$, b is legally *in* w.r.t. *Lab* and φ.

The deductive support relation

Definition 5.4.11 (Legal label w.r.t. one deductive support)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F such that there is a deductive support $\varphi \in\Rightarrow_d$ with $b \in \mathsf{src}(\varphi)$. We say that

- b is *legally in w.r.t. Lab and* φ iff some element of $\{\varphi\} \cup (\mathsf{src}(\varphi) \setminus \{b\})$ has the acceptance label *out* w.r.t. *Lab*, or some element of $\mathsf{trg}(\varphi)$ has the acceptance label *in* w.r.t. *Lab*.

- b is *legally out w.r.t. Lab and* φ iff every element of $\{\varphi\} \cup (\mathsf{src}(\varphi) \setminus \{b\})$ has the acceptance label *in* w.r.t. *Lab*, and every element of $\mathsf{trg}(\varphi)$ has the acceptance label *out* w.r.t. *Lab*.

Definition 5.4.12 (Legal label w.r.t. deductive supports)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F. We say that

- b is *legally in w.r.t. Lab and deductive supports* iff for every deductive support $\varphi \in\Rightarrow_d$ with $b \in \mathsf{src}(\varphi)$, b is legally *in* w.r.t. *Lab* and φ.

- b is *legally out w.r.t. Lab and deductive supports* iff for some deductive support $\varphi \in\Rightarrow_d$ with $b \in \mathsf{src}(\varphi)$, b is legally *out* w.r.t. *Lab* and φ.

The necessary support relation

Definition 5.4.13 (Legal label w.r.t. one necessary support)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F such that there is a necessary support $\varphi \in\Rightarrow_n$ with $b \in \mathsf{trg}(\varphi)$. We say that

- b is *legally in w.r.t. Lab and* φ iff some element of $\{\varphi\} \cup (\mathsf{trg}(\varphi) \setminus \{b\})$ has the acceptance label *out* w.r.t. *Lab*, or some element of $\mathsf{src}(\varphi)$ has the acceptance label *in* w.r.t. *Lab*.

- b is *legally out w.r.t. Lab and* φ iff every element of $\{\varphi\} \cup (\mathsf{trg}(\varphi) \setminus \{b\})$ has the acceptance label *in* w.r.t. *Lab*, and every element of $\mathsf{src}(\varphi)$ has the acceptance label *out* w.r.t. *Lab*.

Definition 5.4.14 (Legal label w.r.t. necessary supports)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F. We say that

- b is *legally in w.r.t. Lab and necessary supports* iff for every necessary support $\varphi \in\Rightarrow_n$ with $b \in \mathsf{src}(\varphi)$, b is legally *in* w.r.t. *Lab* and φ.

- b is *legally out w.r.t. Lab and necessary supports* iff for some necessary support $\varphi \in \Rightarrow_n$ with $b \in \mathsf{src}(\varphi)$, b is legally *out* w.r.t. *Lab* and φ.

Combining all relations

Definition 5.4.15 (Legal label)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F, and let b be an element of F. We say that

- b is *legally in w.r.t. Lab* iff b is legally *in* w.r.t. *Lab* and attacks, incompatibilities, deductive supports and necessary supports and every element of $\mathsf{pre}(b)$ has acceptance label *in* w.r.t. *Lab*.

- b is *legally out w.r.t. Lab* iff b is legally *out* w.r.t. *Lab* and either attacks, deductive supports or necessary supports, or some element of $\mathsf{pre}(b)$ has acceptance label *out* w.r.t. *Lab*.

- b is *legally undec w.r.t. Lab* iff b is neither legally *in* nor legally *out* w.r.t. *Lab*.

Definition 5.4.16 (Admissible EEAF labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F. *Lab* is called an *admissible labeling* of F iff the following conditions hold:

- For every $x \in NonEx(F)$, if the acceptance label of x w.r.t. *Lab* is *in*, then x is legally *in* w.r.t. *Lab*.

- For every $x \in NonEx(F)$, if the acceptance label of x w.r.t. *Lab* is *out*, then x is legally *out* w.r.t. *Lab*.

- For every $(\varphi, x) \in PES(F)$, if the explanatory label of (φ, x) is `exp`, then (φ, x) is legally `exp` w.r.t. *Lab*.

- For every $(\varphi, x) \in PES(F)$, if the explanatory label of (φ, x) is `nexp`, then (φ, x) is legally `nexp` w.r.t. *Lab*.

Definition 5.4.17 (Complete EEAF Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $Lab = (Lab_{NonEx}, Lab_{PES})$ be a labeling of F. *Lab* is called a *complete labeling* of F iff *Lab* is an admissible labeling of F and the following additional condition holds:

- For every $x \in NonEx(F)$, if the acceptance label of x w.r.t. *Lab* is *undec*, then x is legally *undec* w.r.t. *Lab*.

AC-labelings and EC-labelings of EEAFs are now defined in a similar way as for EAFS in Section 5.2.

An *argumentative core labeling* is defined to be a complete labeling with a maximal set of *in*-labeled arguments and a maximal set of explained explananda:

Definition 5.4.18 (EEAF AC-labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $Lab = (Lab_{NonEx}, Lab_{PES})$ a labeling of F. We say that *Lab* is an *argumentative core labeling (AC-labeling)* of F iff *Lab* is a complete labeling of F and there is no complete labeling $Lab' = (Lab'_{NonEx}, Lab'_{PES})$ of F such that $\{b \in NonEx(F) \mid Lab'_{NonEx}(b) = in\} \supsetneq \{b \in$

$\mathit{NonEx}(F) \mid \mathit{Lab}_{\mathit{NonEx}}(b) = \mathit{in}\}$ or $\{e \in \mathcal{X} \mid \text{for some } b \in \mathit{NonEx}(F), \mathit{Lab}'_{\mathit{PES}}((b,e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathit{NonEx}(F),$
$\mathit{Lab}_{\mathit{PES}}((b,e)) = \texttt{exp}\}$.

For defining the *explanatory core labeling*, we first need the notion of *having explanatory relevance*, which captures the idea of an exp-labeled potential explanation step that contributes to the explanation of an explanandum through a path of exp-labeled potential explanation steps:

Definition 5.4.19 (EEAF Explanatory Relevance)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, let $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ be a labeling of F, and let $(\varphi, y) \in \mathit{PES}(F)$. We say that (φ, y) *has explanatory relevance w.r.t. Lab* iff $\mathit{Lab}_{\mathit{PES}}((\varphi, y)) = \texttt{exp}$ and there is a path $(y_0, \ldots, y_n)$ such that $y_0 = y$, $y_n \in \mathcal{X}$ and for every $0 \leq i < n$, $(y_i, y_{i+1}) \in \mathit{PES}$ and $\mathit{Lab}_{\mathit{PES}}((y_i, y_{i+1})) = \texttt{exp}$.

We then adapt the notion of a satisfactory labeling from EAFs to EEAFs. The formulation is adapted to the labeling semantics for EEAFs, but the intuition is the same, namely that satisfactory labelings are ones which are admissible and maximize explanatory power, i.e. the set of explananda for which it provides an explanation.

Definition 5.4.20 (EEAF Satifactory Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ is a *satisfactory labeling* of F iff *Lab* is an admissible labeling of F and there is no admissible labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathit{NonEx}}, \mathit{Lab}'_{\mathit{PES}})$ of F such that $\{e \in \mathcal{X} \mid \text{for some } b \in \mathit{NonEx}(F), \mathit{Lab}'_{\mathit{PES}}((b,e)) = \texttt{exp}\} \supsetneq \{e \in \mathcal{X} \mid \text{for some } b \in \mathit{NonEx}(F), \mathit{Lab}_{\mathit{PES}}((b,e)) = \texttt{exp}\}$.

Insightful labelings for EEAFs are defined in a similar manner from satisfactory labelings. Instead of maximizing depth, here we maximize explanatory relevance. The only difference is that we have labels on the targets of explanatory relation directly, instead of looking at arguments linked by explanations.

Definition 5.4.21 (EEAF Insightful Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ is an *insightful labeling* of F iff *Lab* is a satisfactory labeling of F and there is no satisfactory labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathit{NonEx}}, \mathit{Lab}'_{\mathit{PES}})$ of F such that $\{(\varphi, y) \in \mathit{PES}(F) \mid (\varphi, y)$ *has explanatory relevance w.r.t. Lab*$'\} \supsetneq \{(\varphi, y) \in \mathit{PES}(F) \mid (\varphi, y)$ *has explanatory relevance w.r.t. Lab*$\}$.

Finally, EC-labelings for EEAFs can be derived in the same manner, namely as insightful labelings which minimize the set of *in* non-explanatory elements.

Definition 5.4.22 (EEAF EC-Labeling)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ a labeling of F. We say that $\mathit{Lab} = (\mathit{Lab}_{\mathit{NonEx}}, \mathit{Lab}_{\mathit{PES}})$ is an *explanatory core labeling (EC-labeling)* of F iff *Lab* is an insightful labeling of F and there is no insightful labeling $\mathit{Lab}' = (\mathit{Lab}'_{\mathit{NonEx}}, \mathit{Lab}'_{\mathit{PES}})$ of F such that $\{b \in \mathit{NonEx}(F) \mid \mathit{Lab}'_{\mathit{NonEx}}(b) = \mathit{in}\} \subsetneq \{b \in \mathit{NonEx}(F) \mid \mathit{Lab}_{\mathit{NonEx}}(b) = \mathit{in}\}$.

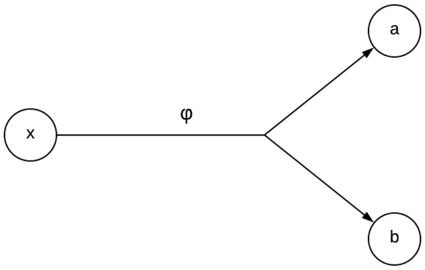


Figure 5.1: Example disjunctive attack.

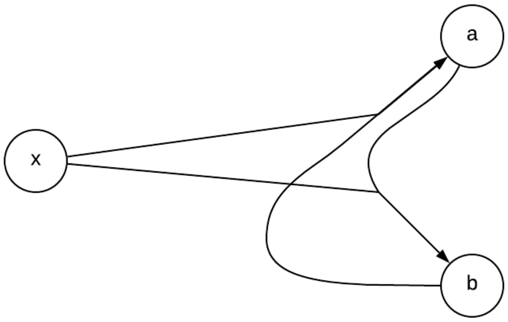


Figure 5.2: Semi-flattened disjunctive attack from Fig. 5.1.

5.4.2 Flattening attacks

In this subsection, we shall describe how the flattening function handles the attack relation. In [24], they present a flattening for attacks from and to non-empty sets of arguments. We shall adapt their flattening to allow for the empty set as a source of an attack, as well as briefly explain how it behaves when elements from the source or target set are not arguments, or the attack is itself involved in another relation.

If two arguments jointly attack a single one, then the behavior will be similar to the one found in [24], namely that the attack is successful if and only if both attacking arguments are accepted. Let us now observe the behavior when an argument disjunctively attacks two other arguments.

In the case where an argument x disjunctively attacks arguments a and b via an attack named φ, as depicted in Fig. 5.1, Gabbay argues that the acceptance of x results in the rejection of a or b. This is taken in a logical sense such that one could reject a, reject b, or reject both, and thus resulting in 3 possible extensions: $\{x, a\}$, $\{x, b\}$ and $\{x\}$. However, in our mindset of maximizing the sets of acceptable arguments in the AC-extensions, we dismiss the possibility for the disjunctive attack of x to be successful on both a and b simultaneously, and thus only allow for 2 extensions in this scenario, $\{x, a\}$ and $\{x, b\}$. As a consequence of this design decision, the flattening also becomes slightly simpler than the one fully described by Gabbay [24]. This interpretation of the attack from and to sets of elements is also in line with the one given by Nielsen and Parsons in their work on set attacks [59].

As an intermediate step in the flattening, one could interpret the disjunctive attack from x to $\{a, b\}$ as two joint attacks, one from $\{x, a\}$ to b and another from $\{x, b\}$ to a, as depicted in Fig. 5.2. This follows the intuition that as long as x is *in*, making either a or b *in* activates a joint attack on the other argument, forcing it *out*.

Observe the flattened framework in Fig. 5.3. The added node φ represents the attack from the original framework, and thus any other relation involving φ in the original framework will, in the flattened version, interact with φ. First notice that we have an auxiliary argument representing the complement of x, $-x$, which will be *out* when x is *in*, *in* when x is *out* and *undec* when x is *undec*. This allows for the attack φ to be shut down in case x

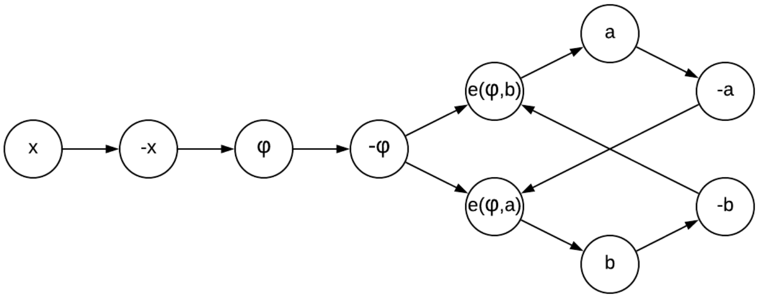


Figure 5.3: Fully-flattened disjunctive attack from Fig. 5.1.

is *out*, by making φ *out* and thus its own complement $-\varphi$ *in*, which will in turn make both $e(\varphi, a)$ and $e(\varphi, b)$ *out* and therefore allow both a and b to be *in*. A similar outcome will occur when φ is successfully attacked directly, while still allowing x to be *in*. In the case where neither x nor φ are *out*, $-\varphi$ will be *out* and therefore we will have an uninterrupted 6-cycle involving a and b. If a is *in*, then its complement $-a$ will be *out* and so $e(\varphi, a)$ will be *in*, forcing b to be *out*. By symmetry, if b is *in* then a will be *out*.

Let us now have a look at the general case, when several elements join forces to disjunctively attack another set of elements. Consider the scenario in Fig. 5.4. Its full flattening is depicted in Fig. 5.5. We represented the elements in the source and target sets as arguments for graphical simplicity, but each could be any kind of valid element from the framework. Also, $n \geq 0$ and $m \geq 1$. For every element in the source, we have the flattened element attacking a complement auxiliary argument, which then attacks the main argument φ. This last argument is the one that represents the attack, so that the acceptance status of φ in the original framework is the same as the acceptance status of this argument in the flattened framework, and any other relation with φ in its source or target set in the original framework will interact with this argument in the flattened version. This argument, and every argument in the target set then also each have a complement which they attack. We then also create coalition arguments in a manner similar to the one originating from the source, following the intuition that a disjunctive attack can be represented as multiple joint attacks. For every element ψ in the target set, all other elements join with φ to attack ψ. The complement of every element also attacks every coalition that the element appears in, so that if an element is *out*, its complement will be *in* and will shut down any coalitions it participated in.

We now formally define the flattening function for the general case. Note that, for simplicity, we will directly use the original elements themselves to represent them in the flattened framework. Since we sometimes introduce quite a few auxiliary arguments in the flattening process, this also has the advantage of making clear which of the added arguments represents the original element.

Definition 5.4.23 (Flattening Attacks)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\varphi \in \rightarrow$. We define $\mathsf{aux}(\varphi) = \{-\psi \mid \psi \in \{\varphi\} \cup \mathsf{src}(\varphi) \cup \mathsf{trg}(\varphi)\} \cup \{e(\varphi, \mathsf{trg}(\varphi) \setminus \{\psi\}) \mid \psi \in \mathsf{trg}(\varphi)\}$. We then define a local flattening function f_l as $f_l(F, \varphi) = \langle \mathcal{A}', \emptyset, \rightarrow', \emptyset, \emptyset \rangle$, such that:

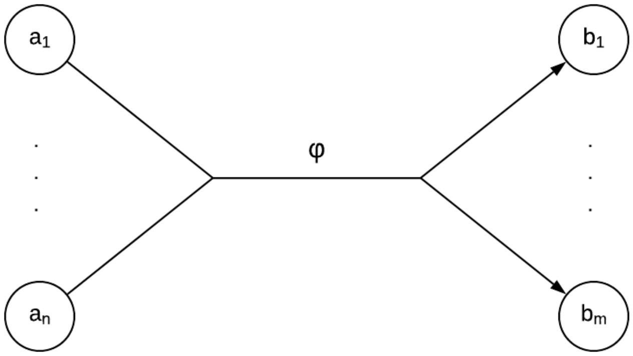


Figure 5.4: Set attacking a set via an attack φ.

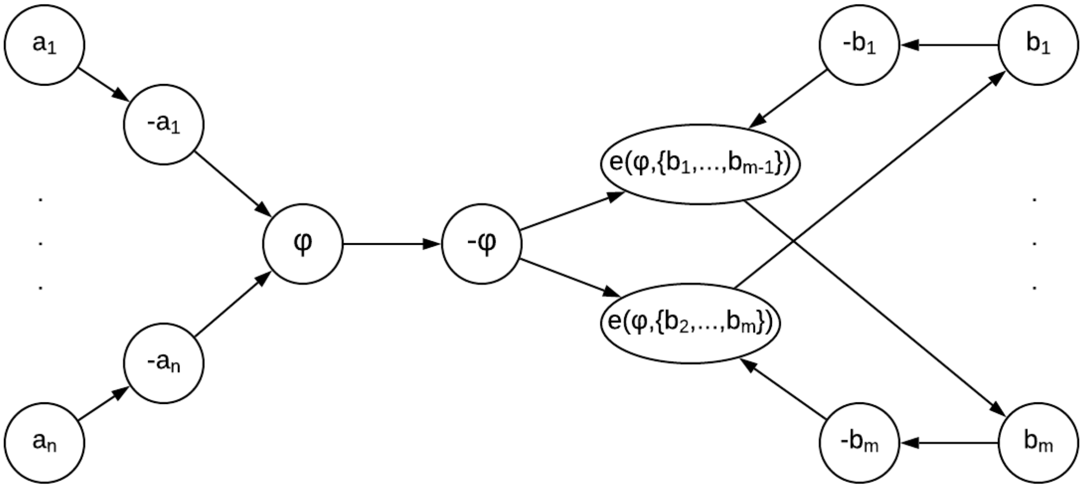

a_1
$-a_1$
$-b_1$
b_1
$e(\varphi,\{b_1,\ldots,b_{m-1}\})$
φ
-φ
$e(\varphi,\{b_2,\ldots,b_m\})$
$-a_n$
a_n
$-b_m$
b_m

Figure 5.5: Flattened attack from Fig. 5.4.

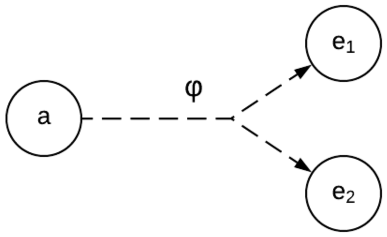


Figure 5.6: Example disjunctive explanation on two explananda.

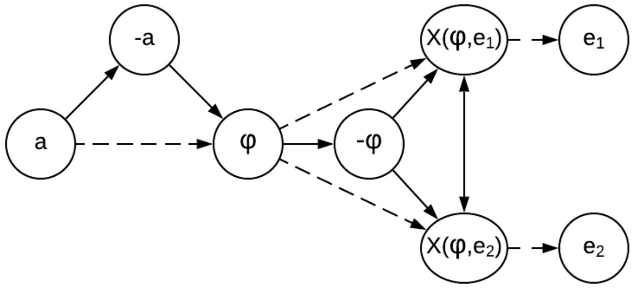


Figure 5.7: Flattening of the disjunctive explanation depicted in Fig. 5.6.

- $\mathcal{A}' = \{\varphi\} \cup \mathsf{src}(\varphi) \cup \mathsf{trg}(\varphi) \cup \mathsf{aux}(\varphi)$;
- $\rightarrow' = \{(\psi, -\psi) \mid \psi \in \{\varphi\} \cup \mathsf{src}(\varphi) \cup \mathsf{trg}(\varphi)\} \cup \{(-\psi, \varphi) \mid \psi \in \mathsf{src}(\varphi)\} \cup \{(-\varphi, e(\varphi, \mathsf{trg}(\varphi) \setminus \{\psi\})) \mid \psi \in \mathsf{trg}(\varphi)\} \cup \{(-\psi, e(\varphi, \mathsf{trg}(\varphi) \setminus \{\chi\})) \mid \psi, \chi \in \mathsf{trg}(\varphi), \psi \neq \chi\} \cup \{(e(\varphi, \mathsf{trg}(\varphi) \setminus \{\psi\}), \psi) \mid \psi \in trg(\varphi)\}$.

Note that when the source of an attack φ is the empty set, we shall simply not have any auxiliary arguments attacking φ, so that the only way to defend from such an attack is to attack φ directly.

5.4.3 Flattening explanations

In this subsection, we now focus on the explanation relation, which we generalize to allow for sets of arbitrary elements as sources and targets of the relation. The interpretation is similar to the one for the attack relation, so that if a set is the source of an explanation, it is only accepted if all elements of the set are *in*, and if a set is the target of an explanation, then the explanation is only successful on a single one of the elements from that set.

Consider a case where an argument disjunctively explains two explananda, as depicted in Fig. 5.6. An example for such a situation could be in the case where a theory is based on the disjunction of two incompatible assumptions, so that in general the consequences of this theory are the same, except when considering its relation to the two explananda. When one of the assumptions holds, then the theory explains one explananda, while when the other assumption holds, the theory explains the other explananda. This scenario is flattened as depicted in Fig. 5.7. $X(\varphi, e_1)$ and $X(\varphi, e_2)$ attack each other, ensuring that only one of the explananda is provided an explanation for. The explanatory link from a to the explananda is preserved by having it explain the right auxiliary arguments. This link is either broken when the relation itself is *out*, or when the corresponding $X(\varphi, e_i)$ is *out*. The complement arguments are also there to ensure that if the argument or the relation is *out*, the rest of the interaction is shut down.

We now observe the general case, when a set of elements explains another set of elements, as depicted in Fig. 5.8. Its flattening is represented in Fig. 5.9. On the side of the

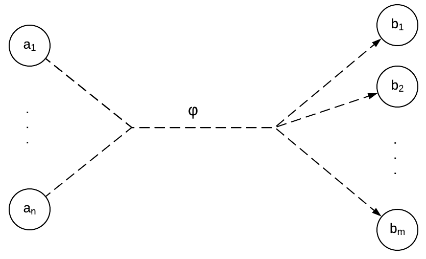


Figure 5.8: General case of explanation by a set of elements and of a set of elements.

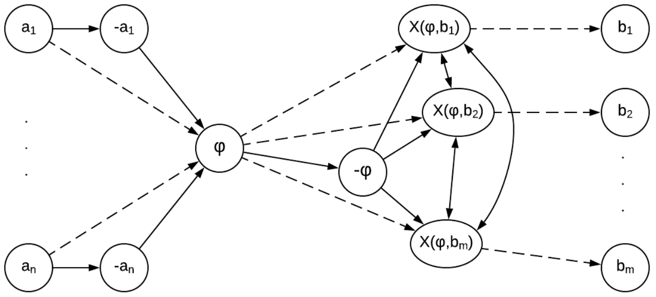


Figure 5.9: Flattening of the general case of explanation depicted in Fig. 5.8.

sources, we have that every element of the source explains the relation φ, giving the possibility to indirectly explain one of the targets. We additionally have that the complement of every source element attacks φ, so that unless all source elements are *in*, at least one complement will be *in* and will make φ *out*, preventing the relation from taking effect. On the side of the targets, we have that for every element being explained by the relation, there is an $X(\varphi, b_i)$ auxiliary argument that explains it. However, all of these $X(\varphi, b_i)$ attack each other, so that only one of them will be *in* and therefore only one of the targets will be successfully explained in the end. This is however not the case when φ is *out*, since we then have that $-\varphi$ is *in* and successfully attacks all $X(\varphi, b_i)$, preventing any of the targets from being explained by φ.

We formally define the flattening depicted in Fig. 5.9:

Definition 5.4.24 (Flattening Explanations)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\varphi \in \dashrightarrow$. We define a local flattening function f_l as $f_l(F, \varphi) = \langle \mathcal{A}', \mathcal{X}', \rightarrow', \dashrightarrow', \emptyset \rangle$, such that:

- $\mathcal{A}' = \{\psi, -\psi \mid \psi \in \{\varphi\} \cup \mathsf{src}(\varphi)\} \cup \{\psi, X(\varphi, \psi) \mid \psi \in \mathsf{trg}(\varphi) \cap \mathcal{A}\}$;
- $\mathcal{X}' = \{\psi, X(\varphi, \psi) \mid \psi \in \mathsf{trg}(\varphi) \cap \mathcal{X}\}$;
- $\rightarrow' = \{(\psi, -\psi) \mid \psi \in \{\varphi\} \cup \mathsf{src}(\varphi)\} \cup \{(-\psi, \varphi) \mid \psi \in \mathsf{src}(\varphi)\} \cup \{(-\varphi, X(\varphi, \psi)) \mid \psi \in \mathsf{trg}(\varphi)\} \cup \{(X(\varphi, \psi), X(\varphi, \chi)) \mid \psi, \chi \in \mathsf{trg}(\varphi), \psi \neq \chi\}$;
- $\dashrightarrow' = \{(\psi, \varphi) \mid \psi \in \mathsf{src}(\varphi)\} \cup \{(\varphi, X(\varphi, \psi)) \mid \psi \in \mathsf{trg}(\varphi)\} \cup \{(X(\varphi, \psi), \psi) \mid \psi \in \mathsf{trg}(\varphi)\}$.

5.4.4 Flattening incompatibility

In this subsection, we focus on the incompatibility relation, which we generalize to allow for an arbitrary non-empty set of elements to be consindered an incompatible set of element. The intuitive interpretation of an incompatibility of a set of elements is that not all elements of the set can be accepted together. Just like in the case of the binary incompatibility relation in EAFs, the incompatibility relation cannot be used to defend an argument.

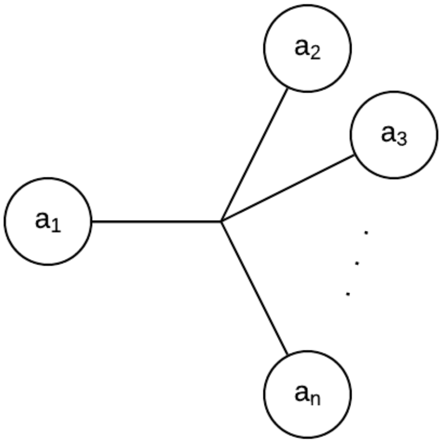


Figure 5.10: Depiction of the incompatibility of the set of elements $\{a_1, \dots, a_n\}$.

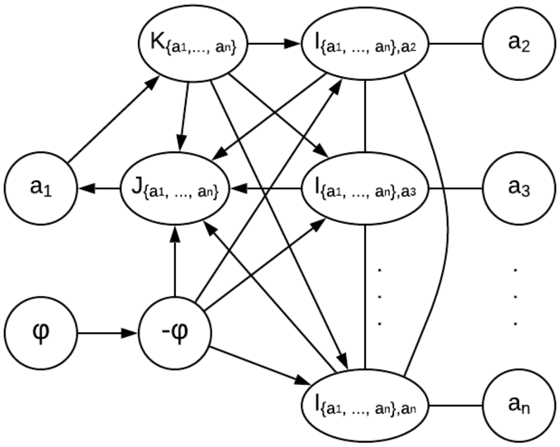


Figure 5.11: Flattening of the general case of incompatibility depicted in Fig. 5.10.

We depict the incompatibility of a set $\{a_1, \dots, a_n\}$ of elements as shown in Figure 5.10:

Let us first illustrate how the incompatibility relation is flattened when the incompatible set of argument contains two arguments, say a and b. Intuitively, as long as the element representing the incompatibility of this set is not the target of a relation of the EEAF, this case should behave the same way as the binary incompatibility between a and b behaves in an EAF. However, we cannot just flatten this set incompatibility on $\{a, b\}$ to the binary incompatibility between a and b, beacuase this would not allow us to correctly treat the case when the element representing the set incompatibility is the target of a relation of the EEAF. Instead, we flatten it as shown in Figure 5.13:

At first sight it might be surprising that we flatten a symmetric relation between two elements in this asymmetric way. But of course the important point is not whether the flattening is asymmetric from a purely syntactic point of view, but whether its semantic behavior is symmetric and coincides with the intended semantic behavior. This is indeed the case for this flattening.

Example 5.4.1. Consider the case with three incompatible arguments depicted in Fig. 5.14 and flattened in Fig. 5.15. Supposing there is no external attack on any of the elements, for a to be *in*, we need $J_{\{a,b,c\}}$ to be *out*. So, we need either $I_{\{a,b,c\},b}$ or $I_{\{a,b,c\},c}$ to be *in*. This means that we cannot have both b and c be *in*, but as soon as exactly one of them is *in*, a becomes *in* as well. Similarly, for b to be *in*, we need $I_{\{a,b,c\},b}$ to not be *in*. So we need either $K_{\{a,b,c\}}$ to be *in*, which requires a to be *out* and also makes c *in*, or $I_{\{a,b,c\},c}$ to be *in*, which forces c to not be *in* and makes a *in*. So in all cases, with no outside influence, the only extensions contains exactly two of the three arguments a, b and c. Interested readers can also verify that even with other relations interacting with the incompatible arguments, it does not matter which argument is chosen as the lone argument (a in this example).

Figures 5.15 and 5.11 show how a 3-element set and a general n-element set get flat-

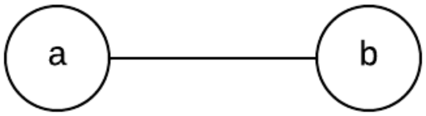


Figure 5.12: Example incompatibility between two arguments.

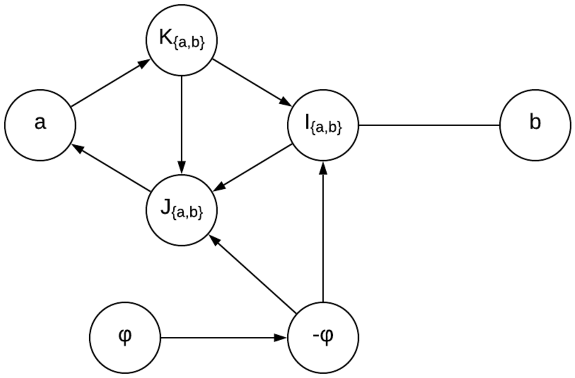


Figure 5.13: Flattening of the incompatibility between two arguments depicted in Figure 5.12.

tened:

We now formally define the flattening function for the general case. Note that we require the Axiom of Choice, in order to be able to select one element from the incompatible ones as the odd one out, due to the asymmetrical construction of the flattening. So given a possibly infinite set of elements Θ, we assume a function $\textsf{choose}$ which returns exactly one element from Θ. It does not matter which element is chosen, however one is required for the construction.

Formally speaking, an incompatibility is a set of elements from the framework. However, to avoid potential confusion and improve readability, for an incompatibility φ, we write φ when referring to the incompatibility itself and $\textsf{src}(\varphi)$ when referring to the set of incompatible elements, even though they are formally the same objects.

Definition 5.4.25 (Flattening Incompatibility). Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, $\varphi \in \sim$ and let $\psi = \textsf{choose}(\varphi)$. We define $\textsf{aux}(\varphi) = \{-\varphi, J_{\textsf{src}(\varphi)}, K_{\textsf{src}(\varphi)}\} \cup \{I_{\textsf{src}(\varphi),\psi} \mid \psi \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi))\}\}$. We then define a local flattening function f_l as $f_l(F, \varphi) = \langle \mathcal{A}', \emptyset, \rightarrow', \emptyset, \sim' \rangle$, such that:

- $\mathcal{A}' = \{\varphi\} \cup \textsf{src}(\varphi) \cup \textsf{aux}(\varphi)$;
- $\rightarrow' = \{(\varphi, -\varphi), (-\varphi, J_{\textsf{src}(\varphi)}), (K_{\textsf{src}(\varphi)}, J_{\textsf{src}(\varphi)}), (J_{\textsf{src}(\varphi)}, \textsf{choose}(\textsf{src}(\varphi))), (\textsf{choose}(\textsf{src}(\varphi)), K_{\textsf{src}(\varphi)})\} \cup \{(-\varphi, I_{\textsf{src}(\varphi),\psi}) \mid \psi \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi))\} \cup \{(K_{\textsf{src}(\varphi)}, I_{\textsf{src}(\varphi),\psi}) \mid \psi \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi))\} \cup \{(I_{\textsf{src}(\varphi),\psi}, J_{\textsf{src}(\varphi)}) \mid \psi \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi))\}$;
- $\sim' = \{(\psi, I_{\textsf{src}(\varphi),\psi}), (I_{\textsf{src}(\varphi),\psi}, \psi) \mid \psi \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi))\} \cup \{(I_{\textsf{src}(\varphi),\psi}, I_{\textsf{src}(\varphi),\psi'}), (I_{\textsf{src}(\varphi),\psi'}, I_{\textsf{src}(\varphi),\psi}) \mid \psi, \psi' \in \textsf{src}(\varphi) \setminus \{\textsf{choose}(\textsf{src}(\varphi)), \psi \neq \psi'\}$.

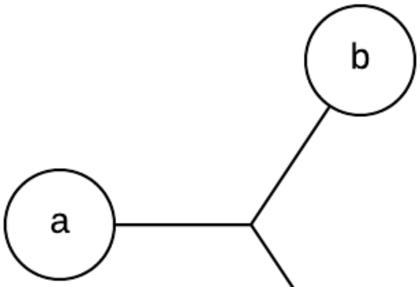


Figure 5.14: Example incompatibility between three arguments.

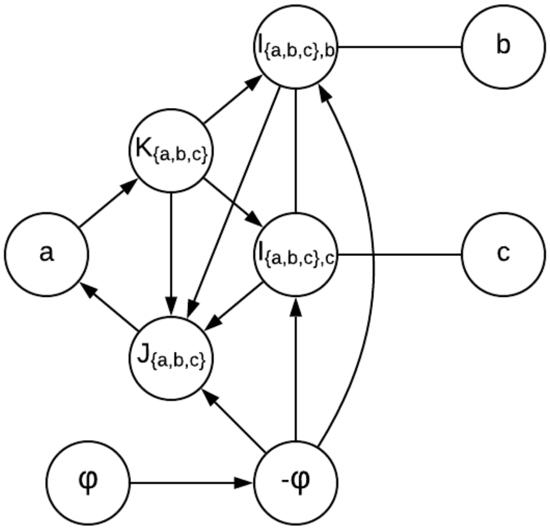


Figure 5.15: Flattening of the incompatibility between three arguments depicted in Fig. 5.14.

5.4.5 Flattening necessary and deductive support

In this subsection, we now focus on the support relations, both the necessary and the deductive ones. In both cases, which we generalize them to allow for sets of arbitrary elements as sources and targets of the relation.

For necessary support, the interpretation is as follows: when a source set necessarily supports a target set, if no element of the source is accepted and the support relation is not rejected, then no element in the target set can be accepted either.

Consider a case where a set of two arguments necessary supports two other arguments, as depicted in Fig. 5.16. For a concrete example illustrating the behavior, picture a chef improvising a new dish. He may opt for a sweet dish, such as a cake, or a savory dish, such as a green salad. If he wishes his dish to be contain notes of both however, he needs a good reason to do so, as these tastes usually do not go well together. In that case, many reasons are valid, and any of them alone would be enough to justify his choice to include both, but without any acceptable reason it seems questionable to include both.

Example 5.4.2. A formal example is depicted in Fig. 5.16. Here, we have a set of two arguments, $\{a, b\}$, necessary supporting another set of two arguments, $\{c, d\}$. We introduce a semi-flattening here, which still includes a disjunctive attack, in order to help the reader with the intuition behind the full flattening. So the semi-flattening goes as follows: the two arguments from the source set, a and b, both attack an auxiliary argument x_φ, so that as long as one of either a or b is accepted, this auxiliary argument x_φ is rejected. This argument is additionally attacked by the complement of the relation argument, $-\varphi$, which is itself attacked by φ. This leads to the result that if φ is *out* then its complement $-\varphi$ will be *in*, making x_φ *out* and therefore canceling any impact the status of the source argument would

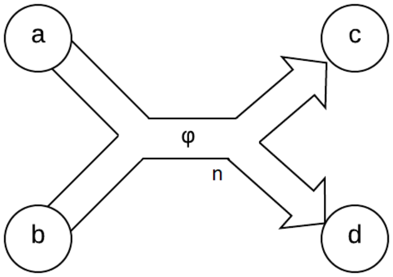


Figure 5.16: Example of a set of two arguments necessary supporting another set of two arguments.

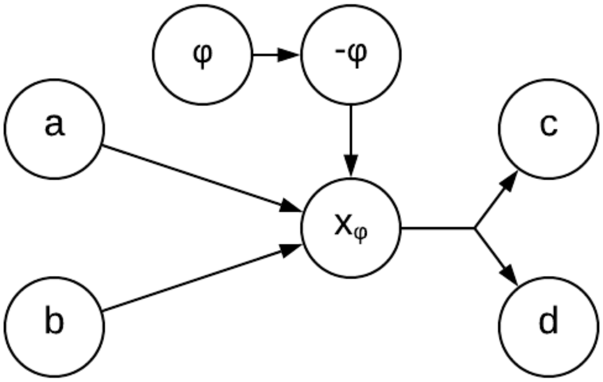


Figure 5.17: Semi-flattening of the set support depicted in Fig. 5.16.

otherwise have on the status of the target arguments. In case both of the source arguments are *out* and the relation argument φ is *in*, the auxiliary argument x_φ will be *in* and therefore the set $\{c, d\}$ is disjunctively attacked, meaning that at least one of c or d must be *out*. This corresponds to the intuition given previously: since neither of the necessary conditions are acceptable and the relation itself is not contested, it cannot be that the target set is wholly accepted.

The semi-flattening depicted in Fig. 5.17 still contains a disjunctive attack, and is therefore not fully flattened into an EAF. When flattening that disjunctive attack as defined earlier, one obtains a full flattening which we now formally define as follows:

Definition 5.4.26. Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\varphi \in \Rightarrow_n$. We define a local flattening function f_l as $f_l(F, \varphi) = \langle \mathcal{A}', \emptyset, \rightarrow', \emptyset, \emptyset \rangle$, such that:

- $\mathcal{A}' = \{\varphi, -\varphi\} \cup \mathsf{src}(\varphi) \cup \{x_\varphi, -x_\varphi\} \cup \{\psi, -\psi \mid \psi \in \mathsf{trg}(\varphi)\} \cup \{e(x_\varphi, \mathsf{trg}(\varphi) \setminus \psi) \mid \psi \in \mathsf{trg}(\varphi)\}$;
- $\rightarrow' = \{(-\varphi, x_\varphi)\} \cup \{(\psi, x_\varphi) \mid \psi \in \mathsf{src}(\varphi)\} \cup \{(\psi, -\psi) \mid \psi \in \mathsf{trg}(\varphi) \cup \{\varphi, x_\varphi\}\} \cup \{(-\psi, e(x_\varphi, \mathsf{trg}(\varphi) \setminus \chi)) \mid \psi \in \mathsf{trg}(\varphi) \cup \{-x_\varphi\}, \chi \in \mathsf{trg}(\varphi), \psi \neq \chi\} \cup \{(e(x_\varphi, \mathsf{trg}(\varphi) \setminus \psi), \psi) \mid \psi \in \mathsf{trg}(\varphi)\}$.

We now focus on the case of deductive support. The interpretation is simply the converse of necessary support: if no elements in the target set is accepted but the support relation is, then no element in the source set can be accepted.

Consider a case where a set of two arguments deductively supports two other arguments, as depicted in Fig. 5.18. For a concrete example illustrating the behavior, picture the following scenario: a university professor has 2 PhD students, each of whom has a paper accepted at the famous conference X. We now consider two arguments: first, a: "the remaining travel budget is Y", and second, b: "sending a PhD student to conference X costs

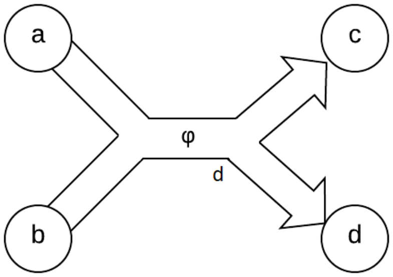


Figure 5.18: Example of a set of two arguments deductively supporting another set of two arguments.

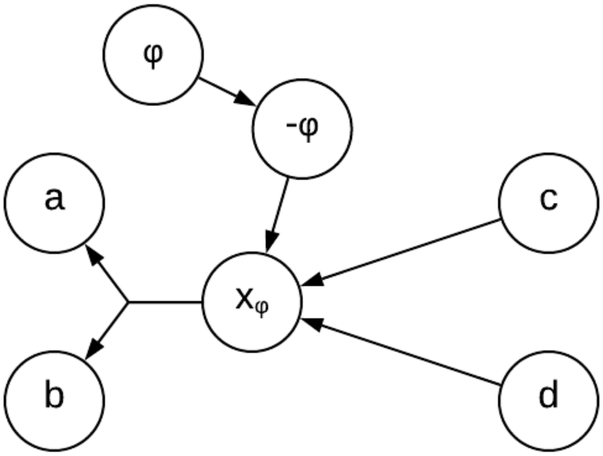


Figure 5.19: Semi-flattening of the set support depicted in Fig. 5.18.

exactly Y”. Additionally, we have two more arguments: c: ‘student 1 cannot go to conference X”, and d: “student 2 cannot go to conference X”. Now, one can see that together, the arguments a and b jointly imply that either c or d holds, since the professor cannot send both students to the conference. So as long as one accepts both the arguments a and b, one must be prepared to accepted at least one of c or d. This means that if one does not accept neither c nor d, one cannot simultaneously accept a and b.

Example 5.4.3. A formal example is depicted in Fig. 5.18, with its flattening in Fig. 5.19. The flattening is essentially the same as for necessary support, but starting from the target and going towards the source.

Similarly as for the case of necessary support, the semi-flattening depicted in Fig. 5.19 still contains a disjunctive attack, and is therefore not fully flattened into an EAF. When flattening that disjunctive attack as defined earlier, one obtains a full flattening which we now formally define as follows:

Definition 5.4.27. Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF and $\varphi \in \Rightarrow_d$. We define a local flattening function f_l as $f_l(F, \varphi) = \langle \mathcal{A}', \emptyset, \rightarrow', \emptyset, \emptyset \rangle$, such that:

- $\mathcal{A}' = \{\varphi, -\varphi\} \cup \mathsf{trg}(\varphi) \cup \{x_\varphi, -x_\varphi\} \cup \{\psi, -\psi \mid \psi \in \mathsf{src}(\varphi)\} \cup \{e(x_\varphi, \mathsf{src}(\varphi) \setminus \psi) \mid \psi \in \mathsf{src}(\varphi)\}$;
- $\rightarrow' = \{(-\varphi, x_\varphi)\} \cup \{(\psi, x_\varphi) \mid \psi \in \mathsf{trg}(\varphi)\} \cup \{(\psi, -\psi) \mid \psi \in \mathsf{src}(\varphi) \cup \{\varphi, x_\varphi\}\} \cup \{(-\psi, e(x_\varphi, \mathsf{src}(\varphi) \setminus \chi)) \mid \psi \in \mathsf{src}(\varphi) \cup \{-x_\varphi\}, \chi \in \mathsf{src}(\varphi), \psi \neq \chi\} \cup \{(e(x_\varphi, \mathsf{src}(\varphi) \setminus \psi), \psi) \mid \psi \in \mathsf{src}(\varphi)\}$.

5.4.6 Combining the flattenings

Now that we have defined the local flattening of each relation, we define how these flattenings are combined to fully flatten any EEAF into an EAF.

Given an EAF $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$, we define $arg(F) = \mathcal{A}$, $explananda(F) = \mathcal{X}$, $att(F) = \rightarrow$, $explanation(F) = \dashrightarrow$ and $incomp(F) = \sim$.

Definition 5.4.28. Given a set of EAFs S, we define the *union* of these EAFs as $\bigcup S = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim \rangle$ where:

- $\mathcal{A} = \bigcup \{arg(F) \mid F \in S\}$
- $\mathcal{X} = \bigcup \{explananda(F) \mid F \in S\}$
- $\rightarrow = \bigcup \{att(F) \mid F \in S\}$
- $\dashrightarrow = \bigcup \{explanation(F) \mid F \in S\}$
- $\sim = \bigcup \{incomp(F) \mid F \in S\}$

For the global flattening, we simply locally flatten every single relation in the EEAF, and then take the union of the resulting EAFs.

Definition 5.4.29. Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF. We define the *global flattening* function f_g as $f_g(F) = \bigcup \{f_l(F, \varphi) \mid \varphi \in (\mathit{NonEx}(F) \setminus \mathcal{A})\}$.

We show that the local flattening preserves the acceptability status of the non-auxiliary elements.

For the sake of conciseness, we say that an element e of an EAF or EEAF F is *legally labeled* w.r.t. a labeling *Lab* of F if $\mathit{Lab}(e) = \mathit{in}$ and e is legally *in* w.r.t. *Lab*, $\mathit{Lab}(e) = \mathit{out}$ and e is legally *out* w.r.t. *Lab*, or $\mathit{Lab}(e) = \mathit{undec}$ and e is legally *undec* w.r.t. *Lab*.

Lemma 5.4.1. *Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow, \sim, \Rightarrow_d, \Rightarrow_n \rangle$ be an EEAF, $a \in \mathit{NonEx}(F)$, $\varphi \in \rightarrow$ and Lab a labeling of F. If a is legally labelled with respect to Lab and attack φ, then there exists a labeling Lab$'$ of the flattened EAF F' where for all $\psi \in \mathit{NonEx}(F)$, $\mathit{Lab}(\psi) = \mathit{Lab}'(\psi)$, and every element in $\mathsf{aux}(\varphi)$ is legally labeled w.r.t. Lab$'$.*

5.5 Related research

The work of Oren and Norman [31] introduces a notion of evidential support. Here, the arguments' attacks are effective only if the attacking arguments are backed by evidence, either directly or indirectly via a chain of support from other arguments. Here, the evidence takes the form of support from the environment itself, in other words facts originating from the context. A similar behavior could be achieved in our framework with support from the empty set. Although the intuition is similar, it remains to be seen how closely related the formal aspects are.

Gottifredi et al. [60] present a framework with recursive attacks and necessary support. An earlier version [61] proposed to derive the semantics from a translation, similar to the flattening approach described in this chapter. The more recent work instead introduces direct semantics which do not require the translation. An avenue of research for future work would be to investigate whether one could derive direct semantics for EEAFs in a similar fashion, obtaining an extension-based semantics equivalent to both the labelling and flattening approaches for EEAFs.

In a recent study, Flouris and Bikakis [62] investigate the framework with sets of attacking arguments (SETAF). This framework adds joint attacks to the base abstract argumentation frameworks, allowing non-empty sets of arguments to attack single arguments. This could be seen as a restriction of EEAFs where the explanations, explananda, supports and incompatibilities are empty, and no joint attack is made on any non-singleton set of arguments. In their work, they present adaptation of existing semantics for abstract argumentation frameworks to SETAFs, together with a labelling semantics for each of those semantics. Due to the important role explanations play for EEAF semantics, such adaptations seem less obvious. It could however be fruitful to investigate how one could adapt well-known abstract argumentation semantics to EEAFs where the explanations, explananda and incompatibilities are empty.

5.6 Applying EEAFs to self-reference paradoxes

Let us now move on to some examples, which focus on solutions for logical paradoxes of self-reference. The arguments are extracted from three different excerpts of *Saving Truth from Paradox* [58].

In the first example, two groups of solutions to the liar paradox are compared. The first group is the solutions which weaken classical logic, namely the paracomplete, paraconsistent and semi-classical solutions. The second group is comprised of the underspill and overspill solutions.

We have the following arguments:

- E_p: This explanandum represents the paradox.
- A: The paracomplete, paraconsistent and semi-classical solutions which provide explanations for the paradox by weakening classical logic.
- B: The underspill and overspill solutions which provide their own explanation of the paradox by suggesting that for some predicates F, F is true of some objects that aren't F or vice-versa.
- C: We did not change logic to hide the defects in other flawed theories such as Ptolemaic astronomy, so why should we change the logic simply to hide these paradoxes?
- D: There is no known way of saving these flawed theories such as Ptolemaic astronomy and even if there was, there is little benefit to doing so.
- F: We have worked out the details of the new logics and they allow us to conserve the theory of truth.
- G: Changing the logic implies changing the meaning.

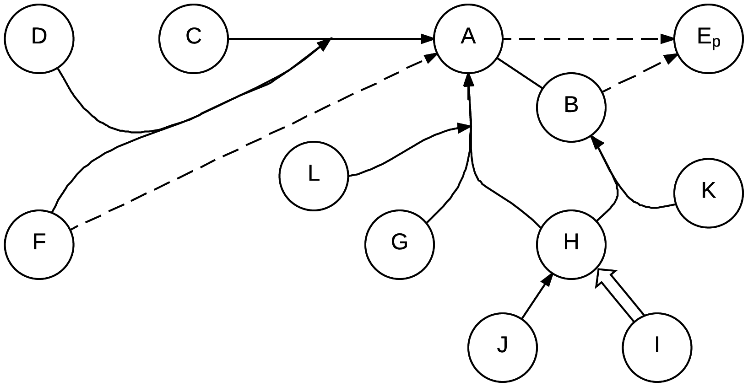


Figure 5.20: EEAF representing the reasoning behind the first excerpt

- H: Change of meaning is bad.
- I: The change is mere.
- J: This is no 'mere' relabelling.
- K: Change of truth schema is a change of the meaning of 'true'.
- L: The paradox forces a change of meaning.

The framework is represented in Figure 5.20 and its flattening in Figure 5.21. We have omitted less-relevant auxiliary arguments for the sake of visibility.

We get that the AC-extensions are $\{A, C, D, F, L, G, J, K\}$ and $\{B, C, D, F, L, G, J, K\}$. We can distinguish here the two rivaling solutions which are both selected. This is due to the fact that even though the author might have a preference for one or another, in the excerpt we have analyzed, he is merely defending the solutions represented in A from attacks and making no argument which attacks the solutions represented in B.

The EC-extensions are $\{A, D, F, L\}$, $\{A, D, F, J\}$ and $\{B, J\}$. Notice that there are two different EC-extensions which contain A, as there are two arguments which individually defend A from the coalition attack of $\{G, H\}$.

The second example focuses on the Russell property. There is a principle which the author refers to as (INST), where F is some intelligible predicate:

(INST) "The property of being F is instantiated by all and only those things that are F."

Now the Russell property is the property of not instantiating itself. By plugging the Russell property in INST, we get:

"The Russell property is instantiated by all and only those things that don't instantiate themselves."

Which can also be rephrased as:

"The Russell property instantiates itself if and only if it does not instantiate itself."

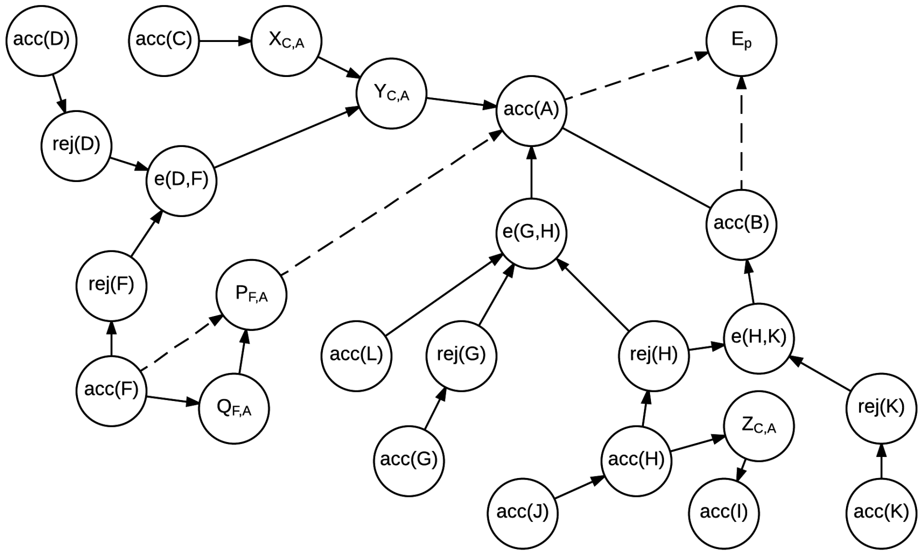


Figure 5.21: Flattened EEAF representing the reasoning behind the first excerpt

This is the Russell paradox, which is similar in many ways to the liar paradox as it has the form $B \leftrightarrow \neg B$. Let us now examine the following arguments:

- E_p: this is an explanandum which represents Russell's paradox.
- A: The Non-existence solution, which suggests that there is no such property as the Russell property.
- B: One could argue that it would violate the raison d'être of properties to suppose that for an intelligible predicate such as 'doesn't instantiate itself', there is no corresponding property of not instantiating itself.
- C: As an answer to this, one could deny that the Russell property is intelligible.
- D: It seems odd to say that the property of not instantiating itself is not intelligible as all parts of it are intelligible.
- F: By defining intelligible as "expresses a property", one can deny that the Russell property is intelligible.

These arguments give us the framework in Fig. 5.22.

The non-existence solution A explains the paradox E_p and is attacked by the argument B that it violates the raison dêtre of properties to suppose that such a property does not exist. B is in turn attacked by the argument C that the property is not intelligible, which also deepens A's explanation. The argument D then states that all parts of 'does not instantiates itself' are intelligible and thus attacks C. We then have the argument F that 'intelligible should be read as 'expresses a property". This attacks D and also adds to the explanatory depth of C as it explains the term 'intelligible' used in C. However, notice that F also

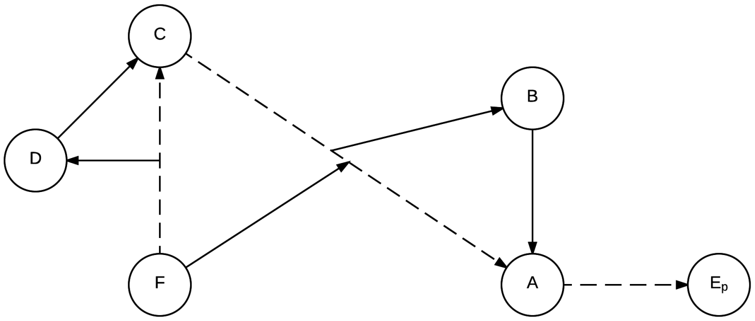


Figure 5.22: EEAF representing the reasoning behind the second excerpt

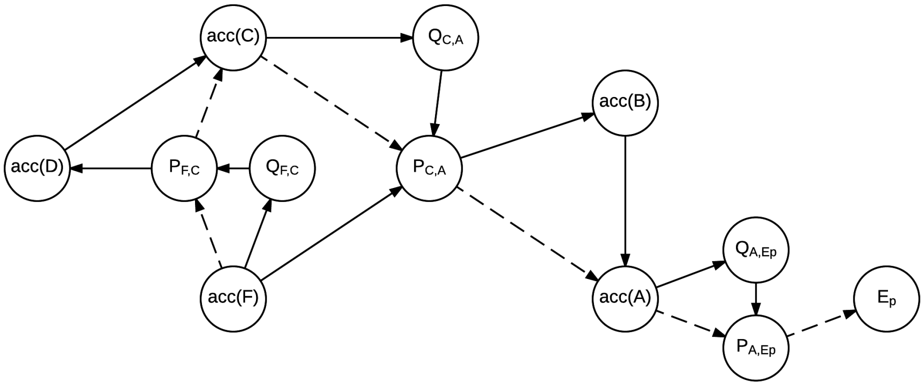


Figure 5.23: Flattened EEAF representing the reasoning behind the second excerpt

attacks the explanatory relation from C to A as the definition of 'intelligible' and 'express a property' given by the solution are now defined in terms of each other and can thus never be fully settled. This means that the solution has not explained the failure of the property (COMP). As a consequence, F also defends B from the attack based on C's explanation of A.

The framework is flattened as depicted in Fig. 5.23 (we have again omitted less-relevant auxiliary arguments for the sake of visibility).

We get that the only AC-extension is $\{B, C, F\}$. On the other hand, the only EC-extensions is the empty-set. This is due to the fact that the only argument explaining the explanandum E_p is A, yet A is attacked by B which is defended by the unattacked argument F. Hence, A can never be defended and thus we can extract no relevant explanation from this framework.

Let us now examine the last set of arguments. Here, the author is focusing on the paracomplete solutions to the liar paradox. The paracomplete solutions reject the principle of excluded middle which states that for every formula φ, it always holds that $\varphi \vee \neg\varphi$. We have the following arguments:

- E_p: This explanandum represents once again the paradox.

- A: The paracomplete solution explains the paradox by rejecting the law of excluded middle.

- B: Why would one reject the law of excluded middle when it seems sound in mathematics, physics etc.

- C: The paracomplete solution only question its applicability to certain circular predicates such as this paradox.

- D: An interesting paracomplete theory in which the Naive Property Theory is consistent might not even be possible since intuitionist logic invalidates the central argument from equivalence to contradiction but still allows for contradictions from a formula such as $B \leftrightarrow \neg B$.

- F: In deMorgan logics without LEM, $B \leftrightarrow \neg B$ is not contradictory.

- G: $B \leftrightarrow \neg B$ not being contradictory is not enough, we also need to maintain Naive Property Theory and include intersubstitutivity of equivalents.

- H: Intersubstitutivity of equivalents follows from (INST) in classical logic.

- I: We are considering logics weaker than classical logic in which it may not follow from (INST).

- J: In the reasonably strong deMorgan logic advocated later in the book, (INST) holds.

We get the framework in Fig. 5.24, which gets flattened and simplified into the framework from Fig. 5.25.

The only AC-extension is $\{A, C, F, G, H, I, J\}$ while the EC-extension is $\{A, C, F, J\}$. This means that the solution modeled here is consistent with the proposed arguments F, G, H and I, even if they do not directly contribute to the explanation of the solution. In the end, the solution A is defended by C from B and by F from D, which is then defended by J from G. Hence, the four arguments A, C, F, J are essential and sufficient to defend the solution in this model. Note that no arguments are explaining each other, hence no measure of explanatory depth is performed in this model.

5.7 Conclusion and further research

We have examined several extensions of abstract argumentation frameworks that add explanatory features, recursive attacks, support and joint attacks. In the cases of recursive attacks, support and joint attacks, we have presented a flattening function, which allows us to instantiate these extended framework as standard AFs. We have shown that in the case of AFRAs, the complete semantics defined in terms of the flattening is equivalent to the complete semantics which has been defined directly on AFRAs. We have then aggregated

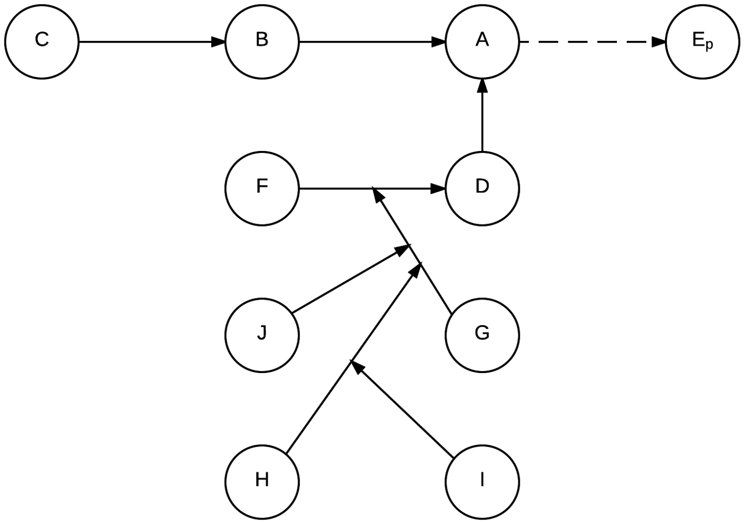


Figure 5.24: EEAF representing the reasoning behind the third excerpt

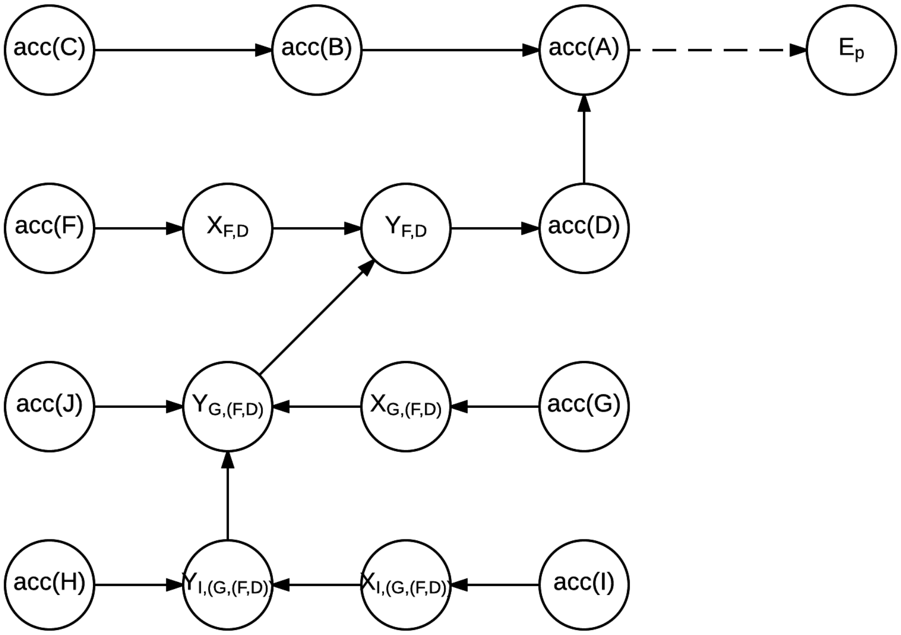


Figure 5.25: Flattened EEAF representing the reasoning behind the third excerpt

these extensions into one framework, EEAFs, and defined the semantics in terms of its flattening to EAFs. Finally, we have explored an application of EEAFs to argumentation from a research-level philosophy book.

Concerning future work in the line of research of this chapter, we plan to extend the result about the flattening of AFRAs to other argumentation semantics than the complete semantics. Furthermore, it might be interesting to investigate flattening the explanatory relation and explananda. Due to their intricate nature, it is not obvious how to flatten them and obtain semantics equivalent to the ones defined on EAFs. Another point of interest would be to apply EEAFs to other areas of scientific debates and examine whether the current features provide enough expressive power.

Chapter 6

Structured argumentation with hypothetical reasoning

6.1 Introduction

Scientific research in the formal sciences (mathematics, logic, theoretical computer science, axiomatic metaphysics, formal linguistics, game theory etc) comes in multiple degrees of formality: fully formal work, which is often performed with the help of computer systems for interactive theorem proving, as it quickly becomes too tedious for humans to explicate all their reasoning in a formal system; fully rigorous proofs that practitioners precisely know how to formalize; practically rigorous work that practitioner know to be formalizable in principle; and informal work like rough proof sketches and considerations about the advantages and disadvantages of various formal systems. Historically, there has been a move from more informal approaches to more formal ones, e.g. in the mathematics of the first half of the 20th century, when the foundational crisis led to an increased attention to axiomatization and to rigorous proofs. This move has generally been accompanied by debates among formal scientists, e.g. about the acceptability of foundational principles and proposed axiomatizations. Despite being informed by formal considerations, these debates have generally been articulated in an informal or semi-formal way.

In this chapter, we propose to use the methodology of *structured argumentation theory* [63] to produce formal models of such informal and semi-formal debates in the formal sciences. Structured argumentation theory allows for a fine-grained model of argumentation and argumentative reasoning based on a formal language and evaluated according to the principles developed in Dung-style *abstract argumentation theory* [4, 64].

One of the dominant formal frameworks for structured argumentation is the *ASPIC+ framework* [6]. In ASPIC+, arguments are built from axioms and premises as well as from strict and defeasible rules, in a similar manner as proofs are built from axioms and rules in a Hilbert-style proof system. The distinction between strict and defeasible rules amounts to the difference between deductively valid modes of inference (e.g. conjunction introduction), and defeasible principles that generally hold but allow for exception (e.g. that dogs generally have four legs). Three kinds of attacks between arguments, *undermines*, *undercuts* and *rebuttals*, are defined between arguments, and finally an *argumentation semantics* from abstract argumentation theory [64] is applied to determine which sets of arguments can be rationally accepted.

Arguments in the formal sciences often involve hypothetical reasoning, which involves reasoning based on an assumption or hypothesis that is locally assumed to be true for the sake of the argument, but to which there is no commitment on the global level. Such hypothetical reasoning is captured well by natural deduction proof systems, whereas the Hilbert-style definition of arguments in ASPIC+ cannot account for such hypothetical reasoning.

ASPIC+ does not allow strict rules to be attacked, which means that debates about which rules of inference are correct, cannot be modeled in ASPIC+. But sometimes formal scientists debate about which rules of inference are deductively valid. ASPIC-END replaces the strict rules of ASPIC+ by *intuitively strict rules*, which formalize the *prima facie* laws of logic which we pre-theoretically consider to be valid without exceptions, but which can nevertheless be given up after more careful examination. Unlike the strict rules of ASPIC+, an intuitively strict rule can be attacked by another argument, but unlike for a defeasible rule, the conclusion of an intuitive strict rule cannot be rejected if both the antecedent of the rule and the rule itself is accepted.

Scientific discourse is characterized not only by the exchange of arguments in favor and against various scientific hypotheses, but also by the attempt to provide scientific *explanations*. In the context of abstract argumentation, [27] have therefore proposed to incorporate the notion of *explanation* into argumentation theory, in order to model scientific debate more faithfully. So far, this incorporation of explanation into argumentation theory has not been extended to the case of structured argumentation. The two contributions of the current chapter in this direction are a general framework for incorporating explanation into structured argumentation and a particular proposal for how to define explanations in instantiations of that framework in the domain of paradoxes arising in the formal sciences.

We propose an adaptation of the ASPIC+ framework called *ASPIC-END* that allows for incorporating hypothetical reasoning and explanations (see Section 6.3). We illustrate the applicability of the framework to debates in the formal sciences through two instantiations of the framework: First, we present in detail a model of a very simple set of arguments about proposed solutions to the Liar Paradox (see Section 6.4). The presentation of this model only serves to illustrate the functioning of ASPIC-END on a simple example and does not purport to be a model of philosophically noteworthy arguments on this topic. In Section 6.5 we sketch and discuss a more extensive model that formalizes parts of the debate that mathematicians had about the Axiom of Choice in the early 20th century [65]. Given that the model still leaves out many contributions to that debate and additionally simplifies some of the contributions that it does take into account, we consider it to only be a preliminary model that we plan to extend in the future. However, we hope that this more extensive model gives some insight into the strengths and drawbacks of the modeling capacities of ASPIC-END, as well as inspiration for further research into this direction.

In order to ensure that the ASPIC-END framework behaves as one would rationally expect, as was previously done for ASPIC+ [66], we prove multiple rationality postulates about ASPIC-END in Section 6.6.

We see two primary motivations for applying the methodology of structured argumentation theory to debates in the formal sciences: First, it is a suitable testbed for structured argumentation theory: Applying structured argumentation theory to real-life debates is often very challenging, because of many layers of uncertainty and imprecision in the interpretation of most types of debates, caused by ambiguities and vagueness of natural language, by a lack of a formal understanding of the domain of discourse of the debate, as well as by the limited rationality of the humans involved in the debate. In the case of debates in the formal sciences, all of these problems are alleviated to some degree: Formal scientists tend to avoid ambiguities and minimize vagueness in their scientific usage of natural language, especially so in the more formal parts of their work, but also in the more informal parts. We have a much better formal understanding of the domains of discourse of the formal sciences than of practically any other domains of discourse. And the debates that scientists have on scientific topics of their field generally show a higher degree of rationality than debates that non-scientists have. For these reasons, it can be hoped that structured argumentation theory can be more easily, and thus hopefully more fruitfully, applied to debates in the formal sciences than to many other kinds of debates to which it has been applied so far. This could also more clearly than existing application bring to light the drawbacks of current approaches in structured argumentation theory, which could become an impetus for further developments in the field.

The second motivation for applying structured argumentation theory to debates in the formal sciences is that in the long run, once the methodology and the models it produces become more mature, such models could contribute to a better understanding of what is at stake in debates in the formal sciences, and hence to a better understanding about the foundations of formal sciences. In this respect, we see the proposed methodology as complimentary to and combinable with the work within the emerging field of *computation metaphysics*, in which methods from automated and interactive theorem proving are used to fully formalize axiomatic theories of metaphysics. The term *computation metaphysics* was first coined by [67], who formalized parts of Abstract Object Theory [68] with PROVER9. More recently, significant contributions to this field of research were made by [69], who with the help of an automated higher-order theorem prover discovered a so far undetected inconsistency in Gödel's ontological argument, and by [70], who used higher-order theorem provers to expose some mistakes and novel insights in a long-standing controversy between Háyek and Anderson concerning a variant of Gödel's ontological argument. This work shows that full formalization of work in a formal field of research can yield real benefits to advance the research in such a field. But so far, this methodology has been limited to the study of the object level of formal axiomatic theories, whereas the meta-level debates that formal scientists have about such theories could not be captured within the formalizations. One way in which the methodology proposed in this chapter could complement the existing methodology of automated theorem proving is that is could allow such meta-level debates to also be captured within a formal model, so that the discovery of mistakes and new insights with the help of automated theorem proving could be extended to this level.

6.2 Related work & motivation for ASPIC-END

The work of [4] introduced the theory of *abstract argumentation*, in which one models arguments by abstracting away from their internal structure to focus on the relation of conflict between them. This gives rise to the notion of an *argumentation framework*, which formally is just a directed graph, whose informal interpretation is that the vertices stand for arguments and the edges stand for the attack relation between arguments, i.e. the relation between a counterargument and the argument that it counters. Given an argumentation framework, the goal is to select a set of arguments deemed acceptable on the sole basis of the attack relation between the arguments. There are various approaches for making such selections, based on different criteria such as conflict-freeness (i.e. never simultaneously accepting two arguments where one attacks the other), defense (accepting an attacked argument only if you also accept counterarguments to all its attackers), and maximality (which among other things ensures that an unattacked argument will always be accepted). A selection of arguments that are deemed simultaneously acceptable according to some criteria is called an *extension*. Sometimes, especially when there are cycles in the argumentation framework, there might be multiple extensions that satisfy the given criteria. For this reason, the formal definition of an *abstract argumentation semantics* is that it is a function that maps any given argumentation framework to a set of sets of arguments (vertices) of that argumentation framework.

In *structured argumentation*, one models also the internal structure of arguments through a formal language in which arguments and counterarguments can be constructed [63]. One important family of frameworks for structured argumentation is the family of ASPIC-like frameworks, which is based on the work of John Pollock (e.g. [71, 72]) and consists among others of the original ASPIC framework [73], the ASPIC+ framework [6], and the ASPIC-framework [74]. We briefly sketch ASPIC+, as it is the basis for our framework ASPIC-END.

In ASPIC+, one starts with a knowledge base and a set of rules[1] which allow one to make inferences from given knowledge. There are two kinds of rules: *Strict rules* logically entail their conclusion, whereas *defeasible rules* only create a presumption in favour of their conclusion. Arguments are built either by introducing an element of the knowledge base into the framework, or by making an inference based on a rule and the conclusions of previous arguments. Attacks between arguments are constructed either by attacking a fallible premise of an argument (*undermining*), by attacking the conclusion of a defeasible inference made within an argument (*rebuttal*), or by questioning the applicability of such a rule (*undercutting*). Preferences between arguments can be derived from preferences between rules. An abstract argumentation framework can thus be built and acceptable arguments can be selected using any abstract argumentation semantics.

[75] have introduced the notion of *rationality postulates* for structured argumentation frameworks. These are conditions that structured argumentation frameworks would rationally be expected to satisfy, such as closure under strict rules of the output and consistency of the conclusions given consistency of the strict rules. [75] showed that the original ASPIC system did not satisfy these postulates, but proposed minor changes that made it satisfy

[1] In this chapter, we use the word *rule* in the way in which it is usually used in the structured argumentation literature. There is one important difference between this usage of *rule* and the way the word is usually used in the logical literature outside of structured argumentation theory: A *rule*, as the word is used in structured argumentation theory, is what would normally be called an instance of a rule. For this reason, it makes sense to speak of a *rule scheme* (as we will frequently do in Section 6.5), which is what would normally be just called a rule.

them. These changes have been incorporated into ASPIC+ [66].

ASPIC-END features three main differences from ASPIC+. The first is that it allows for arguments to introduce an assumption on which to reason hypothetically, just like in natural deduction. In natural deduction, hypothetical derivations are employed in the inference schemes called $\neg$-Introduction (or *proof by contradiction*), $\supset$-Introduction (we use $\supset$ for the material implication), and $\vee$-Elimination (or *reasoning by cases*). Allowing for the usage of defeasible rules within hypothetical reasoning leads to specific problems that have been studied for the inference scheme of reasoning by cases in a recent paper by [76]. In the current chapter we avoid these problems by not allowing defeasible rules within hypothetical reasoning. However, a conclusion made on the basis of an inference scheme involving hypothetical reasoning may still be incorporated into an argument that uses defeasible rules, so that there is some integration of defeasible and hypothetical reasoning.[2]

The second difference is that ASPIC-END allows for arguments about the correct rules of logical reasoning. In ASPIC+, such arguments cannot be modeled, as the rules of logical reasoning represented by strict rules, and arguments involving only strict rules can never be attacked. Argumentation about the correct rules of logical reasoning is quite common in debates in the formal sciences. For example, our *prima facie* intuitions suggest that it is a law of logic that a sentence that is not true must be false. However, the Kripke-Feferman solution to the Liar paradox [78, 79] suggests that some sentences, such as the Liar sentence, are neither true nor false, since giving them either one of the two truth values leads to a contradiction. This solution is not putting forward an argument against the falsehood of the sentence by rebutting it, nor is it undermining any of the argument's premises. It is undercutting the argument by attacking the inference made from the negation of truth to falsehood.

To allow such arguments about the correct laws of logic to be modeled in ASPIC-END, we replace strict rules by *intuitively strict rules* whose applicability can be questioned, as in the case of defeasible rules in ASPIC+, but which behave like strict rules when their applicability is accepted. This means that conclusions of intuitively strict rules cannot be rebutted, just as for strict rules in ASPIC+. Intuitively strict rules represent *prima facie laws of logic*, i.e. purportedly logical inference rules which make sense at first but are open to debate.

The third difference is that ASPIC-END has a notion of *explanations* additionally to the notion of arguments. This feature is based on the work of [27], who have extended Dung-style abstract argumentation with *explananda* (phenomena that need to be explained) and an *explanatory relation*, which allows arguments to either explain these explananda or deepen another argument's explanation. We provide reminders of definitions this chapter builds upon, even though these can also be found in the preliminaries.

Definition 6.2.1 (Explanatory Argumentation Framework)**.** An explanatory argumentation framework (EAF) is a tuple $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$, where $\mathcal{A}$ is a set of arguments, $\mathcal{X}$ is a set of

[2] The early formalisms of [71] and [72] also allowed for arguments involving hypothetical reasoning. Most of the work in structured argumentation theory that built on this early work of Pollock ignored this type of arguments. In a recent paper, [77] have critically assessed the way hypothetical arguments function in Pollock's formalisms and have identified three problematic features of the formalism in [72]. By not allowing defeasible rules within hypothetical reasoning, we avoid these problematic features.

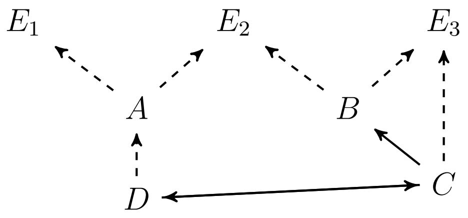


Figure 6.1: Example of explanatory power and depth: $\{B\} >_p \{C\}$ and $\{A, B\} >_p \{B\}$, but $\{A\}$ and $\{C\}$ are incomparable with respect to explanatory power. $\{A, D\} >_d \{A\}$, but $\{A\}$ and $\{B\}$ are incomparable with respect to explanatory depth.

explananda, $\rightarrow$ is an attack relation between arguments and $\dashrightarrow$ is an explanatory relation from arguments to either explananda or arguments.

If $A \dashrightarrow B$, we say that A *explains* B.

Sets of admissible arguments are then selected:

Definition 6.2.2 (EAF Admissible)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be an EAF, $A \in \mathcal{A}$ and $S \subseteq \mathcal{A}$. We say that S is *conflict-free* iff there are no arguments $B, C \in S$ such that $B \rightarrow C$. We say that S *defends* A iff for every $B \in \mathcal{A}$ such that $B \rightarrow A$, there exists $C \in S$ such that $C \rightarrow B$. We say that S is *admissible* iff S is conflict-free and for all $B \in S$, S defends B.

The most suitable admissible sets are then selected by also taking into account their explanatory power and depth. These are measured by first identifying the explanations present in each set of arguments.

Definition 6.2.3 (Explanation Offered)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be an EAF, $S \subseteq \mathcal{A}$ and $E \in \mathcal{X}$. An *explanation* $X[E]$ for E offered by S is a set $S' \subseteq S$ such that there exists a unique argument $A \in S'$ such that $A \dashrightarrow E$ and for all $A' \in S' \setminus \{A\}$, there exists a path in $\dashrightarrow$ from A' to A.

In order to be able to compare sets of arguments on how many explananda they can explain and in how much detail, the two following measures are required:

Definition 6.2.4 (Explanatory Power)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be an EAF and $S, S' \subseteq \mathcal{A}$. Let $\mathcal{E}$ be the set of explananda S offers an explanation for and $\mathcal{E}'$ the set of explananda S' offers an explanation for. We say that S is *explanatory more powerful than* S' ($S >_p S'$) if and only if $\mathcal{E} \supsetneq \mathcal{E}'$.

Definition 6.2.5 (Explanatory Depth)**.** Let $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be an EAF and $S, S' \subseteq \mathcal{A}$. We say that S is *explanatory deeper than* S' ($S >_d S'$) if and only if for each explanation X' offered by S', there is an explanation X offered by S such that $X' \subseteq X$ and for at least one such X and X' pair, $X' \subsetneq X$.

[27] define two procedures for selecting the most suitable sets of arguments. The first procedure (for the *argumentative core*) consists in selecting the most explanatory powerful

conflict-free sets, from which the maximal most defended sets are then retained. The second procedure (for the *explanatory core*) selects the most explanatory powerful conflict-free sets, from which the most defended sets are taken, and then from those selects the minimal explanatory deepest sets. In our formalism, we will slightly alter and reformulate these procedures.

6.3 ASPIC-END

In this section, we define ASPIC-END and motivate the details of its definition.

Definition 6.3.1 (Argumentation Theory)**.** An *argumentation theory* is a tuple $(\mathcal{L}, \mathcal{R}, n, <)$, where:

- $\mathcal{L}$ is a logical language containing a set of free variables $\mathcal{L}_v$ and closed under the binary connective disjunction ($\vee$), the unary connectives negation ($\neg$), the three types of assumability ($Assumable_\neg, Assumable_\vee, Assumable_\supset$), and the existential quantifiers (if $\varphi \in \mathcal{L}$ and $x \in \mathcal{L}_v$, then $\forall x.\varphi, \exists x.\varphi \in \mathcal{L}$) such that $\bot \in \mathcal{L}$.
- $\mathcal{R} = \mathcal{R}_{is} \cup \mathcal{R}_d$ is a set of intuitively strict ($\mathcal{R}_{is}$) and defeasible ($\mathcal{R}_d$) rules of the form $\varphi_1, \ldots, \varphi_n \rightsquigarrow \varphi$ and $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ respectively, where $n \geq 0$ and $\varphi_i, \varphi \in \mathcal{L}$.
- $n : \mathcal{R} \to \mathcal{L}$ is a partial function.
- $\mathcal{R}_{ce} := \{(\bot \rightsquigarrow \alpha) \mid \alpha \in \mathcal{L}\} \subseteq \mathcal{R}_{is}$, and $\forall r \in \mathcal{R}_{ce}$, $n(r)$ is undefined.
- $<$ is an asymmetric and transitive relation over R_d which represents preference.

Note that we interpret $\bot$ not just as any contradiction but as the conjunction of all formulas in the language. We thus require that rules are present in the framework which allow one to derive any formula from $\bot$, which are effectively rules of conjunction elimination.

We now inductively define how to construct arguments. At the same time, we define five functions on arguments that specify certain features of any given argument: $\mathsf{Conc}(A)$ denotes the conclusion of argument A. $\mathsf{As}_\neg(A)$, $\mathsf{As}_\vee(A)$ and $\mathsf{As}_\supset(A)$ denote the set of assumptions under which argument A is operating: $\mathsf{As}_\neg(A)$ stands for the assumptions made for a proof by contradiction, or negation introduction, $\mathsf{As}_\vee(A)$ stands for the assumptions made for reasoning by cases, or disjunction elimination, and $\mathsf{As}_\supset(A)$ stands for the assumptions made for an implication introduction. As a short-hand, we will sometimes write $\mathsf{As}(A) := \mathsf{As}_\neg(A) \cup \mathsf{As}_\vee(A) \cup \mathsf{As}_\supset(A)$. So whenever $\mathsf{As}(A) \neq \emptyset$, A is a hypothetical argument. $\mathsf{Sub}(A)$ denotes the set of sub-arguments of A. $\mathsf{DefRules}(A)$ denotes the set of all defeasible rules used in A. $\mathsf{TopRule}(A)$ denotes the last inference rule which has been used in the argument if such a rule exists, and is undefined otherwise.

Definition 6.3.2 (ASPIC-END Argument)**.** An *argument* A on the basis of an argumentation theory $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ has one of the following forms:

1. $A_1, \ldots, A_n \rightsquigarrow \psi$, where $A_1, \ldots, A_n$ are arguments such that there exists an intuitively strict rule $\mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \rightsquigarrow \psi$ in $\mathcal{R}_{is}$.
   $\mathsf{Conc}(A) := \psi$, $\mathsf{As}_\neg(A) := \mathsf{As}_\neg(A_1) \cup \cdots \cup \mathsf{As}_\neg(A_n)$,
   $\mathsf{As}_\vee(A) := \mathsf{As}_\vee(A_1) \cup \cdots \cup \mathsf{As}_\vee(A_n)$, $\mathsf{As}_\supset(A) := \mathsf{As}_\supset(A_1) \cup \cdots \cup \mathsf{As}_\supset(A_n)$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A_1) \cup \cdots \cup \mathsf{Sub}(A_n) \cup \{A\}$,

$\mathsf{DefRules}(A) := \mathsf{DefRules}(A_1) \cup \cdots \cup \mathsf{DefRules}(A_n)$,
$\mathsf{TopRule}(A) := \mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \rightsquigarrow \psi$.

2. $A_1, \ldots, A_n \Rightarrow \psi$, where $A_1, \ldots, A_n$ are arguments s.t. $\mathsf{As}(A_1) \cup \ldots \cup \mathsf{As}(A_n) = \emptyset$ and there exists a defeasible rule $\mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \Rightarrow \psi$ in $\mathcal{R}_d$.
   $\mathsf{Conc}(A) := \psi$, $\mathsf{As}_\neg(A) := \emptyset$,
   $\mathsf{As}_\vee(A) := \emptyset$, $\mathsf{As}_\supset(A) := \emptyset$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A_1) \cup \cdots \cup \mathsf{Sub}(A_n) \cup \{A\}$,
   $\mathsf{DefRules}(A) := \mathsf{DefRules}(A_1) \cup \cdots \cup \mathsf{DefRules}(A_n) \cup$
   $\{\mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \Rightarrow \psi\}$,
   $\mathsf{TopRule}(A) := \mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \Rightarrow \psi$.

3. $\mathsf{Assume}_\neg(\varphi)$, where $\varphi \in \mathcal{L}$.
   $\mathsf{Conc}(A) := \varphi$, $\mathsf{As}_\neg(A) := \{\varphi\}$,
   $\mathsf{As}_\vee(A) := \emptyset$, $\mathsf{As}_\supset(A) := \emptyset$,
   $\mathsf{Sub}(A) := \{\mathsf{Assume}_\neg(\varphi)\}$,
   $\mathsf{DefRules}(A) := \emptyset$, $\mathsf{TopRule}(A)$ is undefined.

4. $\mathsf{Assume}_\vee(\varphi)$, where $\varphi \in \mathcal{L}$.
   $\mathsf{Conc}(A) := \varphi$, $\mathsf{As}_\neg(A) := \emptyset$,
   $\mathsf{As}_\vee(A) := \{\varphi\}$, $\mathsf{As}_\supset(A) := \emptyset$,
   $\mathsf{Sub}(A) := \{\mathsf{Assume}_\vee(\varphi)\}$,
   $\mathsf{DefRules}(A) := \emptyset$, $\mathsf{TopRule}(A)$ is undefined.

5. $\mathsf{Assume}_\supset(\varphi)$, where $\varphi \in \mathcal{L}$.
   $\mathsf{Conc}(A) := \varphi$, $\mathsf{As}_\neg(A) := \emptyset$,
   $\mathsf{As}_\vee(A) := \emptyset$, $\mathsf{As}_\supset(A) := \{\varphi\}$,
   $\mathsf{Sub}(A) := \{\mathsf{Assume}_\supset(\varphi)\}$,
   $\mathsf{DefRules}(A) := \emptyset$, $\mathsf{TopRule}(A)$ is undefined.

6. $\mathsf{ProofByContrad}(\neg\varphi, A')$, where A' is an argument such that $\varphi \in \mathsf{As}_\neg(A')$ and $\mathsf{Conc}(A') = \bot$.
   $\mathsf{Conc}(A) := \neg\varphi$, $\mathsf{As}_\neg(A) := \mathsf{As}_\neg(A') \setminus \{\varphi\}$,
   $\mathsf{As}_\vee(A) := \mathsf{As}_\vee(A')$, $\mathsf{As}_\supset(A) := \mathsf{As}_\supset(A')$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A') \cup \{\mathsf{ProofByContrad}(\neg\varphi, A')\}$,
   $\mathsf{DefRules}(A) := \mathsf{DefRules}(A')$, $\mathsf{TopRule}(A)$ is undefined.

7. $\mathsf{ReasonByCases}(\psi, A_1, A_2, A_3)$, where:
   A_1 is an argument such that $\varphi \in \mathsf{As}_\vee(A_1)$ and $\mathsf{Conc}(A_1) = \psi$,
   A_2 is an argument such that $\varphi' \in \mathsf{As}_\vee(A_2)$ and $\mathsf{Conc}(A_2) = \psi$,
   A_3 is an argument such that $\mathsf{Conc}(A_3) = \varphi \vee \varphi'$.
   $\mathsf{Conc}(A) := \psi$,
   $\mathsf{As}_\neg(A) := \mathsf{As}_\neg(A_1) \cup \mathsf{As}_\neg(A_2) \cup \mathsf{As}_\neg(A_3)$,
   $\mathsf{As}_\vee(A) := (\mathsf{As}_\vee(A_1) \setminus \{\varphi\}) \cup (\mathsf{As}_\vee(A_2) \setminus \{\varphi'\}) \cup \mathsf{As}_\vee(A_3)$,
   $\mathsf{As}_\supset(A) := \mathsf{As}_\supset(A_1) \cup \mathsf{As}_\supset(A_2) \cup \mathsf{As}_\supset(A_3)$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A_1) \cup \mathsf{Sub}(A_2) \cup \mathsf{Sub}(A_3) \cup \{\mathsf{ReasonByCases}(\psi, A_1, A_2, A_3)\}$,
   $\mathsf{DefRules}(A) := \mathsf{DefRules}(A_1) \cup \mathsf{DefRules}(A_2) \cup \mathsf{DefRules}(A_3)$,
   $\mathsf{TopRule}(A)$ is undefined.

8. $\supset$-intro$(\varphi \supset \psi, A')$, where A' is an argument such that $\varphi \in \mathsf{As}_{\supset}(A')$ and $\mathsf{Conc}(A') = \psi$.
   $\mathsf{Conc}(A) := \varphi \supset \psi$, $\mathsf{As}_{\neg}(A) := \mathsf{As}_{\neg}(A')$,
   $\mathsf{As}_{\vee}(A) := \mathsf{As}_{\vee}(A')$, $\mathsf{As}_{\supset}(A) := \mathsf{As}_{\supset}(A') \setminus \{\varphi\}$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A') \cup \{\supset\text{-intro}(\varphi \supset \psi, A')\}$,
   $\mathsf{DefRules}(A) := \mathsf{DefRules}(A')$, $\mathsf{TopRule}(A)$ is undefined.

9. $\forall$-intro$(\forall x.\varphi(x), A')$, where A' is an argument such that for some $x \in \mathcal{L}_v$, there is no $\psi \in \mathsf{As}(A')$ such that x is free in ψ, and $\mathsf{Conc}(A') = \varphi(x)$.
   $\mathsf{Conc}(A) := \forall x.\varphi(x)$, $\mathsf{As}_{\neg}(A) := \mathsf{As}_{\neg}(A')$,
   $\mathsf{As}_{\vee}(A) := \mathsf{As}_{\vee}(A')$, $\mathsf{As}_{\supset}(A) := \mathsf{As}_{\supset}(A')$,
   $\mathsf{Sub}(A) := \mathsf{Sub}(A') \cup \{\forall\text{-intro}(\forall x.\varphi(x), A')\}$,
   $\mathsf{DefRules}(A) := \mathsf{DefRules}(A')$, $\mathsf{TopRule}(A)$ is undefined.

Notice that we do not allow for the use of defeasible rules within hypothetical arguments, as reflected in the condition of Def. 6.3.2 item 2 that the sub-arguments cannot have any assumptions. We do however allow for the conclusions of defeasible arguments to be imported inside of a hypothetical argument. This is motivated by the fact that allowing for proofs by contradiction amounts to allowing for transpositions of any rule that can be used within a proof by contradiction, and transpositions are usually assumed only for strict rules in structured argumentation [75, 66].

Example 6.3.1. Consider an argumentation theory $AT_1 = (\mathcal{L}, \mathcal{R}, n, <)$, where $\mathcal{L}$ is the smallest set containing $\{p, q, r, s, u\}$ and satisfying Definition 6.3.1 item 1, $\mathcal{R}_{is} = \{p \rightsquigarrow q; q \rightsquigarrow \bot; \rightsquigarrow r\}$, $\mathcal{R}_d = \{\neg p, r \Rightarrow s; u \Rightarrow q\}$ and $<$ is the empty relation. We can then construct an argument for s as follows:

- A_1:= $\mathsf{Assume}_{\neg}(p)$, with $\mathsf{As}_{\neg}(A_1) = \{p\}, \mathsf{Conc}(A_1) = p$
- A_2:= $A_1 \rightsquigarrow q$, with $\mathsf{As}_{\neg}(A_2) = \{p\}, \mathsf{Conc}(A_2) = q$
- A_3:= $A_2 \rightsquigarrow \bot$, with $\mathsf{As}_{\neg}(A_3) = \{p\}, \mathsf{Conc}(A_3) = \bot$
- A_4:= $\mathsf{ProofByContrad}(\neg p, A_3)$, with $\mathsf{As}_{\neg}(A_4) = \emptyset, \mathsf{Conc}(A_4) = \neg p$
- A_5:= $\rightsquigarrow r$, with $\mathsf{As}_{\neg}(A_5) = \emptyset, \mathsf{Conc}(A_5) = r$
- A_6:= $A_4, A_5 \Rightarrow s$, with $\mathsf{As}_{\neg}(A_6) = \emptyset, \mathsf{Conc}(A_6) = s$

We can see that A_1 introduces the assumption p, and from there the arguments A_2 and A_3 manage to derive a contradiction, which allows the construction of argument A_4 with conclusion $\neg p$ under no assumption. We can then use this together with the premise r to form an argument for s. Note however that we cannot form an argument for $\neg u$ using a proof by contradiction, because to derive an inconsistency from u we would have to use d_2. However, defeasible rules can only be applied under no assumption, hence we would be unable to apply it in the proof by contradiction for $\neg u$.

We now need to define the attack relation in our framework. Notice that in ASPIC-END, we also allow for an argument A to attack an argument B which makes an assumption φ if A concludes that φ is not assumable. For example, if one were to assume that the number 5 is yellow, since numbers do not have colors, it should be possible to attack the argument that introduces this assumption and any argument making an inference from this assumption. We also separate the assumption-attack into the three different kinds of assumptions, so that one can, for example, deny a formula's assumability for reasoning by cases but still allow it to be assumed for implication-introduction. Additionally, if one wishes, for example, to refute the well-foundedness of a construction such as proof by contradiction while still accepting reasoning by cases, one simply needs to attack the $\neg$-assumability of all formulas.

Definition 6.3.3 (ASPIC-END Attacks)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory and A, B two arguments on the basis of Σ. We say that A *attacks* B iff A *rebuts*, *undercuts* or *assumption-attacks* B, where:

- A *rebuts* argument B (on B') iff $\mathsf{Conc}(A) = \neg\varphi$ or $\neg\mathsf{Conc}(A) = \varphi$ for some $B' \in \mathsf{Sub}(B)$ of the form $B''_1, \dots, B''_n \Rightarrow \varphi$ and $\mathsf{As}(A) = \emptyset$.
- A *undercuts* argument B (on B') iff $\mathsf{Conc}(A) = \neg n(r)$ or $\neg\mathsf{Conc}(A) = n(r)$ for some $B' \in \mathsf{Sub}(B)$ such that $\mathsf{TopRule}(B') = r$, there is no $\varphi \in \mathsf{As}(B')$ such that $\neg\varphi = \mathsf{Conc}(A')$ or $\varphi = \neg\mathsf{Conc}(A')$ for some $A' \in \mathsf{Sub}(A)$, and there are arguments $B_1, ..., B_n$ such that $B_1 = B'$, $B_n = B$, $B_i \in \mathsf{Sub}(B_{i+1})$ for $1 \leq i < n$ and $\mathsf{As}(A) \subseteq \mathsf{As}(B_1) \cup \dots \cup \mathsf{As}(B_n)$.
- A *assumption-attacks* B (on B') iff for some $B' \in \mathsf{Sub}(B)$ such that $\mathsf{As}(A) = \emptyset$ and one of the following holds:
  - $B' = \mathsf{Assume}_\neg(\varphi)$ and $\mathsf{Conc}(A) = \neg Assumable_\neg(\varphi)$;
  - $B' = \mathsf{Assume}_\vee(\varphi)$ and $\mathsf{Conc}(A) = \neg Assumable_\vee(\varphi)$;
  - $B' = \mathsf{Assume}_\supset(\varphi)$ and $\mathsf{Conc}(A) = \neg Assumable_\supset(\varphi)$.

We require that any attacking argument A is making fewer assumptions than the B' it attacks, as to prevent arguments from attacking outside of their assumption scope. Note that in the case of rebuttal, since the attacked argument cannot have assumptions, we require that the attacking argument have none either.

In the case of undercutting, we also have the requirement that A does not use the contrary of any assumptions made by B' in any of its inferences, since the attack would not stand in the scope of B'. Additionally, we allow A to make use of any assumptions appearing in the chain of arguments leading B' to B, as these assumptions, even if they have been retracted, still constitute valid grounds on which to form an attack.

Similarly as in ASPIC+, one can also define a notion of successful attack by lifting the preference relation from rules to arguments as follows:

Definition 6.3.4 (Lifting of Preference)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory and A, B be two arguments on the basis of Σ. We define the *lifting of* $<$ *to arguments* $\prec$ to be such that $A \prec B$ iff there exists $r_a \in \mathsf{DefRules}(A)$, such that for all $r_b \in \mathsf{DefRules}(B)$, we have $r_a < r_b$.

Notice that this lifting corresponds to elitist weakest-link as described by [6]. We believe that this ordering is best suited for modeling philosophical and scientific arguments.

We now define what it means for an attack to be successful:

Definition 6.3.5 (Defeats)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, A, B be two arguments on the basis of Σ. We say that A *successfully rebuts* B iff A rebuts B on B' for some argument B' and $A \not< B'$, and that A *defeats* B iff A assumption-attacks, undercuts or successfully rebuts B.

The aim of our system is to generate an EAF as defined in Section 6.2. For this three things need to be specified: A set $\mathcal{X}$ of explananda, a condition under which an argument explains an explanandum, and a condition under which an argument explains another argument. The first two of these three details are domain-specific, and are thus to be specified in an instantiation of the ASPIC-END framework. The third one, on the other hand, should be the same in all domains. The reason for this can be found in the informal clarification that [27] provided for what it means to say that an argument b explains an argument a: "argument b can be used to explain one of the premises of argument a [...] or the link between the premises and the conclusion."

In the context of structured argumentation, this informal clarification can be turned into a formal definition:

Definition 6.3.6 (Explanations)**.** Let A, B be arguments. We say that B *explains* A (on A') iff $A' \in \mathsf{Sub}(A)$, $\mathsf{As}(B) \subseteq \mathsf{As}(A')$ and at least one of the following two cases holds:

- $A' \notin \mathsf{Sub}(B)$ and either $A' = (\rightsquigarrow \mathsf{Conc}(B))$ or $A' = (\Rightarrow \mathsf{Conc}(B))$.
- $\mathsf{Conc}(B) = n(\mathsf{TopRule}(A'))$ and $\nexists B' \in \mathsf{Sub}(B)$ such that $\mathsf{TopRule}(B') = \mathsf{TopRule}(A')$.

Intuitively, the idea behind this definition is that an argument B explains another argument A if B non-trivially concludes one of A's premises or one of the inference rules used by A.

We now have all the elements needed to build an EAF.

Definition 6.3.7 (Corresponding EAF)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory. Let $\mathcal{X}$ be a set of explananda, and let $\mathcal{C}$ be a criterion for determining whether an argument constructed from Σ explains a given explanandum $E \in \mathcal{X}$. The *explanatory argumentation framework* (EAF) *defined by* $(\Sigma, \mathcal{X}, \mathcal{C})$ is a tuple $\langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$, where:

- $\mathcal{A}$ is the set of all arguments that can be constructed from Σ satisfying Definition 6.3.2;
- $(A, B) \in \rightarrow$ iff A defeats B, where $A, B \in \mathcal{A}$;
- $(A, E) \in \dashrightarrow$ iff criterion $\mathcal{C}$ is satisfied with respect to A and E, where $A \in \mathcal{A}$ and $E \in \mathcal{X}$;
- $(A, B) \in \dashrightarrow$ iff A explains B according to Definition 6.3.6, where $A, B \in \mathcal{A}$.

Once such a framework has been generated, we want to be able to extract the most interesting sets of arguments. Such a set should be able to explain as many explananda in as much detail as possible, while being self-consistent and plausible.

We define two kinds of extensions corresponding to the two selection procedures defined by [27]. As suggested in the informal discussion in their paper, we chose to give higher importance to the criterion of defense compared to the criterion of explanatory power. This prevents some absurd theories which manage to explain all explananda but cannot defend themselves against all attacks from beating plausible theories which fail to explain some of the explananda but are sound and fully defended.

Definition 6.3.8 (AC- & EC-extensions). Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ the EAF defined by Σ and $S \subseteq \mathcal{A}$ a set of arguments.

1. We say that S is *satisfactory* iff S is admissible and there is no $S' \subseteq \mathcal{A}$ such that $S' >_p S$ and S' is admissible.
2. We say that S is *insightful* iff S is satisfactory and there is no $S' \subseteq \mathcal{A}$ such that $S' >_d S$ and S' is satisfactory.
3. We say that S is an *argumentative core extension* (*AC-extension*) of F iff S is satisfactory and there is no $S' \supset S$ such that S' is satisfactory.
4. We say that S is an *explanatory core extension* (*EC-extension*) of F iff S is insightful and there is no $S' \subset S$ such that S' is insightful.

The AC-extensions are sets of arguments which represent the theories explaining the most explananda, together with all other compatible beliefs present in the framework. EC-extensions represent the core of those theories and only include the arguments which defend or provide details for them.

We define the conclusions of the arguments in a given extension as follows:

Definition 6.3.9 (Conclusions of an Extension). Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ and S be an extension of F. Then, we define the *conclusions of* S, denoted $\mathsf{Concs}(S)$, to be $\mathsf{Concs}(S) = \{\mathsf{Conc}(A) | A \in S$ s.t. $\mathsf{As}(A) = \emptyset\}$.

6.4 Modelling explanations of semantic paradoxes in ASPIC-END

In this section, we discuss how ASPIC-END can be applied to modeling argumentation about explanations of semantic paradoxes, and illustrate this potential application with a simple example. We start by briefly motivating this application of structured argumentation theory.

Philosophy is an academic discipline in which good argumentative skills are a central part of every student's training. Philosophical texts are often much richer in explicit formulation of arguments than texts from other academic disciplines. For these reasons, we believe that modeling arguments from philosophical textbooks, monographs and papers can be an interesting test case for structured argumentation theory.

Different areas of philosophy vary with respect to how much logical rigor is commonly applied in the presentation of arguments. Even logically rigorous argumentation poses many interesting problems, as the rich literature on abstract and structured argumentation attests. In order to not confound these interesting problems with issues arising from the lack of logical rigor, it is a good idea to concentrate on the study of logically rigorous argumentation. Philosophical logic is an area of logic where logically rigorous arguments abound. One topic that has gained a lot of attention in philosophical logic is the study of semantic paradoxes such as the Liar paradox and Curry's paradox [80, 58]. We therefore use the argumentation about the various explanations of the paradoxes that have been proposed in the philosophical literature as a test case for structured argumentation theory.

In our application of ASPIC-END to argumentation about explanations of semantic paradoxes, the explananda are the paradoxes (i.e. arguments that derive an absurdity under no assumption without using defeasible rules), which other arguments can explain by attacking the said derivation. So we instantiate the set $\mathcal{X}$ of explananda and criterion $\mathcal{C}$ for an explanation of an explanandum by an argument as specified in the following two definitions:

Definition 6.4.1 (Generation of Explananda)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory. For every argument A on the basis of Σ such that $\mathsf{DefRules}(A) = \emptyset, \mathsf{As}(A) = \emptyset$ and $\mathsf{Conc}(A) = \bot$, we stipulate an explanandum E_A, and say that $\mathsf{Source}(E_A) = A$. We define the set $\mathcal{X}$ of explananda based on Σ to be the set of all explananda E_A that we have thus stipulated.

Definition 6.4.2 (Satisfying the Explanation Criterion)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, A an argument and E an explanandum based on Σ. We say that criterion $\mathcal{C}$ is satisfied with respect to A and E iff A defeats $\mathsf{Source}(E)$.

The following example illustrates an application of ASPIC-END to a version of the Liar paradox and two very simple explanations of it:[3]

Example: Define L to be the sentence "L is false". If L is true, i.e. "L is false" is true, then L is false, which is a contradiction. So L is not true, i.e. L is false. So "L is false" is true, i.e. L is true. So we have the contradiction that L is both true and false from no assumption.

A truth-value gap explanation: In this paradox, the only inference steps that are not justified by the laws of classical logic are the steps that involve reasoning about the meaning of "true" and "false". Since classical logic is a well-studied system for formalizing rational reasoning, we should accept it. Thus we need to give up some inference rules based on the meaning of "true" and "false". This can be achieved by giving up the assumption that every sentence is either true or false for problematically self-referential sentences such as L. In

[3] Note that our aim here is not to present a detailed case study of how a debate about a semantic paradox can be formalized in ASPIC-END, but only to illustrate the way ASPIC-END works and could be used for such a case study in future work. For this reason, we restrict ourselves to a simple exposition of the Liar paradox and two very simple explanations of it, a truth-value gap explanation and a paracomplete explanation. See [58] for comprehensive presentations of truth-value gap and paracomplete explanations, besides many others. Additionally note that, for the sake of simplicity, we only include in our model those instances of rules that are actually used in the explanations that we formalize, so we leave out other instances of the general rules (rule schemes) that lie behind these instances. A detailed case study would have to consider what happens when all instances of these rules are included; for this purpose, other paradoxes like Curry's paradox and various revenge versions of the Liar paradox would need to be considered as well, as the instances of these rules applied to the paradoxical sentences from these other paradoxes would be included in the model.

the paradox, this assumption is used when concluding that L is false because L is not true, so this inference should be rejected.

A paracomplete explanation: If we give up some of the natural inference rules that are based on the meaning of "true" and "false", our formalism no longer correctly captures the meaning of "true" and "false", so we should not give up these rules. In order to avoid the paradox, we therefore need to limit some rules of classical logic. This can be achieved by allowing a proof by contradiction based on assumption ϕ only in case the law of excluded middle holds for ϕ, i.e. in case $\phi \vee \neg\phi$. The law of excluded middle should not be accepted for problematically self-referential statements like L, and thus also not for the statement "L is true". So "L is true" cannot be assumed for a proof by contradiction, i.e. the derivation of "L is not true" based on deriving a contradiction from the assumption the L is true is not valid.

We now proceed to the ASPIC-END model of the reasoning and argumentation involved in the paradox and the two explananda. We use T and F to mean *true* and *false* respectively; the other abbreviations we use should be self-explanatory from the context. The rules in our model are such that $\mathcal{R}_{is}$ is the smallest set satisfying Def 6.3.1 item 1 and including the rules listed below. For each intuitively strict rule, we provide either a brief explanation of where the rule comes from, or we refer to the name of the corresponding rule in [58], of which the rule in question is an instance:

$T(L) \rightsquigarrow T(F(L))$	(by definition, as L is defined to mean $F(L)$)
$T(F(L)) \rightsquigarrow F(L);$	(T-Elim)
$T(L), F(L) \rightsquigarrow \bot;$	(a sentence cannot be both true and false)
$\neg T(L) \rightsquigarrow F(L);$	(a sentence that is not true is considered false)
$F(L) \rightsquigarrow T(F(L));$	(T-Introd)
$T(F(L)) \rightsquigarrow T(L);$	(by definition, as L is defined to mean $F(L)$)
$\rightsquigarrow \forall r.(\mathit{used\_in\_paradox}(r)$	$\wedge\, \neg \mathit{T\text{-}F\text{-}rule}(r) \supset r \in \mathit{classical\_logic})$
	(all inference rules that are used in the derivation of the paradox and that are not based on the meaning of "true" and "false" are admissible in classical logic)

The naming function is defined by $n(\neg T(L) \rightsquigarrow F(L)) = r_1$. The set $\mathcal{R}_d$ of defeasible rules is defined as follows:

- $\Rightarrow \mathit{formalizes\_rational\_reasoning}(\mathit{classical\_logic});$
- $\mathit{formalizes\_rational\_reasoning}(\mathit{classical\_logic}) \Rightarrow \mathit{accept}(\mathit{classical\_logic});$
- $\forall r.(\mathit{used\_in\_paradox}(r) \wedge \neg \mathit{T\text{-}F\text{-}rule}(r) \supset r \in \mathit{classical\_logic}),$
  $\mathit{accept}(\mathit{classical\_logic}) \Rightarrow \exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r));$
- $\Rightarrow \mathit{problematically\_self\text{-}referential}(L);$
- $\mathit{problematically\_self\text{-}referential}(L), \exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r)) \Rightarrow \neg r_1;$

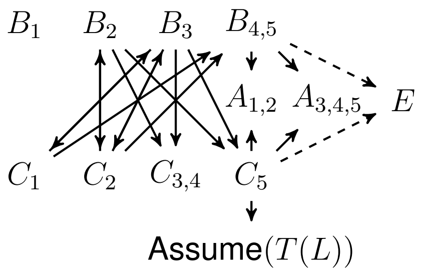


Figure 6.2: The relevant arguments, explanandum, attacks and explanations from the example

- $\Rightarrow \mathit{correctly\_capture}(\mathit{TF\text{-}meaning})$;
- $\mathit{correctly\_capture}(\mathit{TF\text{-}meaning}) \Rightarrow \neg\exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r))$;
- $\forall r.(\mathit{used\_in\_paradox}(r) \wedge \neg\mathit{T\text{-}F\text{-}rule}(r) \supset r \in \mathit{classical\_logic})$,
  $\neg\exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r)) \Rightarrow \neg\mathit{accept}(\mathit{classical\_logic})$;
- $\mathit{problematically\_self\text{-}referential}(L), \mathit{accept}(\mathit{classical\_logic}) \Rightarrow$
  $\neg\mathit{accept}(T(L) \vee \neg T(L))$;
- $\neg\mathit{accept}(T(L) \vee \neg T(L)) \Rightarrow \neg\mathit{Assumable}_{\neg}(T(L))$

Infinitely many arguments can be constructed from this argumentation theory. However, the following set of arguments is the set of most relevant arguments, in the sense that other arguments will not defeat these arguments and will not add relevant new conclusions.

$$\begin{aligned}
A_{1,2} &= \mathsf{ProofByContrad}(\neg T(L), (\mathsf{Assume}_{\neg}(T(L)),\\
&\quad ((\mathsf{Assume}_{\neg}(T(L)) \rightsquigarrow T(F(L))) \rightsquigarrow F(L)) \rightsquigarrow \bot)) \rightsquigarrow F(L)\\
A_{3,4,5} &= ((A_1 \rightsquigarrow T(F(L))) \rightsquigarrow T(L)), A_1 \rightsquigarrow \bot\\
B_1 &= (\rightsquigarrow \forall r.(\mathit{used\_in\_paradox}(r) \wedge \neg\mathit{T\text{-}F\text{-}rule}(r) \supset r \in \mathit{classical\_logic}))\\
B_2 &= B_1, (\Rightarrow \mathit{formalizes\_rational\_reasoning}(\mathit{classical\_logic})) \Rightarrow \mathit{accept}(\mathit{classical\_logic})\\
B_3 &= B_2 \Rightarrow \exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r))\\
B_{4,5} &= (\Rightarrow \mathit{problematically\_self\text{-}referential}(L)), B_3 \Rightarrow \neg r_1\\
C_1 &= (\Rightarrow \mathit{correctly\_capture}(\mathit{TF\text{-}meaning})) \Rightarrow \neg\exists r.(\mathit{T\text{-}F\text{-}rule}(r) \wedge \mathit{give\_up}(r))\\
C_2 &= B_1, C_1 \Rightarrow \neg\mathit{accept}(\mathit{classical\_logic})\\
C_{3,4} &= (\Rightarrow \mathit{problematically\_self\text{-}referential}(L)), C_2 \Rightarrow \neg\mathit{accept}(T(L) \vee \neg T(L))\\
C_5 &= C_{3,4} \Rightarrow \neg\mathit{Assumable}_{\neg}(T(L))
\end{aligned}$$

We get the explanandum E with $\mathsf{Source}(E) = A_{3,4,5}$. $B_{4,5}$ defeats A_2 on A_1 and C_5 defeats A_2 on $\mathsf{Assume}(T(L))$, thus they both explain E. The AC-extensions are $\{B_1, B_2, B_3, B_{4,5}\}$ and $\{B_1, C_1, C_2, C_{3,4}, C_5\}$, and the EC-extensions are $\{B_3, B_{4,5}\}$ and $\{C_2, C_5\}$.

6.5 Modelling argumentation on Axiom of Choice

Additionally to the relatively simple model presented in the previous section, we have also applied ASPIC-END to produce a more extensive model of a debate in the formal sciences, namely a model of parts of the debate that mathematicians had about the Axiom of Choice (AC) in the early 20th century [65]. Given that the author of this thesis contributed only in a small part to the details of this model, we will here only present some fragments of the model, briefly describe some features of the overall model, and discuss some of the the insight into the strengths and drawbacks of the modeling capacities of ASPIC-END that we have gained from producing this model. A complete description of the model can be found in a technical online appendix [81].

In 1904, the German mathematician Ernst Zermelo published a proof of the Well-Ordering Theorem, in which he explicitly referred to a set-theoretic principle that came to be known as the Axiom of Choice [82]. The Axiom of Choice states that for each set M whose elements are non-empty sets, there is a function f that maps each element $m \in M$ to an element $f(m) \in m$. In the first years after its publication, Zermelo's proof received a lot of critique, a significant part of which questioned the general validity of the Axiom of Choice (see [65]). In the long run, however, the proof got accepted, as the Axiom of Choice got accepted as a valid part of the de-facto standard foundational theory for mathematics, *Zermelo-Fraenkel set theory with the Axiom of Choice* (*ZFC*).

The two critiques of Zermelo's Axiom of Choice that we consider in our model are those of [83] and Lebesgue [84]. Furthermore, we consider the counterarguments to these critiques put forward by [85] and by Hadamard [84]. When constructing the formal model, we have made a number of design choices that enabled us to keep the model relatively simple and concise:

- We have only considered the contributions of Zermelo, Peano, Lebesgue and Hadamard to this debate, leaving out some of the other contributions to the debate that are discussed in [65]. The choice of which contributions to include was partially based on the importance of those contributions from the point of view of the history and philosophy of mathematics, and partially based on considerations about which contributions best illustrate the interesting formal features of the ASPIC-END framework.

- In the case of some arguments, we have opted not to formalize the internal details of the argument, but instead include the conclusion of the argument as a defeasible premise in our model, as this significantly simplifies the model. This solution allows the effect of the argument on the overall debate to be faithfully represented even when the internal details of the argument are not made explicit by the model.

- An additional way in which we kept our model simple was by not formalizing in any detail the uncontested mathematical reasoning that is related to the debate, e.g. parts of the proof of the Well-Ordering Principle that do not make use of the Axiom of choice or the proof of the Partition Principle that Zermelo refers to in one of his arguments.

Due to these simplifications, we consider our model to only be a preliminary model that we plan to extend in the future. However, the model already gives some insight into the strengths and drawbacks of the modeling capacities of ASPIC-END, as well as inspiration for further research into this direction.

In our model, the purely mathematical and purely logical demonstrations and reasoning are formalized using intuitively strict rules, while the philosophical and metamathematical argumentation and reasoning is formalized using defeasible rules. Most of the attacks between arguments attack defeasible arguments, i.e. philosophical or metamathematical arguments. But given that some of the mathematical and logical principles that were applied in the mathematical and logical reasoning that we model, e.g. the Axiom of Choice and the non-constructivist parts of classical logic, are attacked by some philosophical or metamathematical arguments, there are also some arguments using only intuitively strict rules that get attacked. By the design of ASPIC-END, all such attacks have to be undercuts.

The debate about the Axiom of Choice that we have formalized in our model concerns the purported justification that Zermelo has given for the Axiom of Choice as well as attacks on this purported justification, but it does not involve any mathematical explanations. For this reason, our model of this debate does not make use of the explanatory machinery included in ASPIC-END, but it does make use of other two novel features of ASPIC-END, i.e. hypothetical reasoning and undercuts of intuitively strict rules.

In order to give a flavor of our formal model, we now present some fragments of it and describe some feature of the overall model. We start by looking at the first argument Zermelo presented for the Axiom of Choice in 1904:

> “this logical principle cannot be reduced to a still simpler one, but is used everywhere in mathematical deduction without hesitation. So for example the general validity of the theorem that the number of subsets into which a set is partitioned is less than or equal to the number of its elements, cannot be demonstrated otherwise than by assigning to each subset one of its elements.” [82, p. 516]

Here are the formal ASPIC-END arguments that we construct to represent this argument and its subarguments:

$$
\begin{aligned}
Z_1^{04} &= (\Rightarrow \text{simple}(AC))\\
Z_2^{04} &= (\Rightarrow \neg\exists x.\ \text{calls\_to\_doubt}(x, \text{usage}(AC)))\\
Z_3^{04} &= (\Rightarrow \exists p.\ \text{demonstrates}(p, PP))\\
Z_4^{04} &= (\Rightarrow \forall p.\ (\text{demonstrates}(p, PP) \supset \text{uses}(p, AC)))\\
Z_5^{04} &= \mathsf{Assume}_{\supset}(\text{demonstrates}(p, PP))\\
Z_6^{04} &= (Z_4^{04}, Z_5^{04} \vdash \exists p, t.\ (\text{demonstrates}(p, t) \wedge \text{uses}(p, AC)))\\
Z_7^{04} &= \supset \textsf{-intro}(\text{demonstrates}(p, PP) \supset \exists p, t.\ (\text{demonstrates}(p, t) \wedge \text{uses}(p, AC)))\\
Z_8^{04} &= \forall\textsf{-intro}(\forall p.\ (\text{demonstrates}(p, PP) \supset \exists p, t.\ (\text{demonstrates}(p, t) \wedge \text{uses}(p, AC))))\\
Z_9^{04} &= (Z_3^{04}, Z_8^{04} \rightsquigarrow \exists p, t.\ (\text{demonstrates}(p, t) \wedge \text{uses}(p, AC)))\\
Z_{10}^{04} &= (Z_9^{06} \Rightarrow \text{widely\_used}(AC))\\
Z_{11}^{04} &= (Z_1^{06}, Z_6^{06}, Z_{10}^{06} \Rightarrow \text{accept}(AC))
\end{aligned}
$$

The rules that are needed to construct these arguments can actually be read off from the arguments, and are explicitly stated in the technical online appendix [81, p. 4-6]. The notation $(A_1, \ldots, A_n \vdash \psi)$ used in argument Z_6^{04} stands for an argument that uses multiple rules of intuitionistic logic to get from the conclusions of arguments A_1, …, A_n to the conclusion ψ. Of course, all these rules are included in our model. Note that argument Z_7^{04} makes use of $\supset$-Introduction, Z_8^{04} makes use of $\forall$-Introduction.

In a letter to Borel that shortly afterwards got published in the Bulletin de la Société mathématique de France [84], Lebesgue made a constructivist argument against the Axiom of Choice:

> "I believe that we can only build solidly by granting that it is impossible to demonstrate the existence of an object without defining it."

We formalize Lebesgue's argument through a defeasible premise according to which an existence proof requires definition and a strict rule that allows to reject the Axiom of Choice based on this defeasible premise:

$$L_1^{05} = (\Rightarrow \text{existence\_proof\_requires\_definition})$$
$$L_2^{05} = (L_1^{05} \rightsquigarrow \neg\text{accept}(AC))$$

The rules that we included in the model in order to formalize the arguments that have been explicitly mentioned in the historical debate on the Axiom of Choice can also be used to construct *implicit arguments* that were not explicitly mentioned in the historical debate. It should not come as a surprise that at the current level of development of our methodology, the model has not given rise to philosophically insightful implicit arguments. However, there is an implicit argument that plays an important role with respect to the formal behavior of our model: It is an argument that makes use of the proof by contradiction to construct an attack on Lebesgue's argument L_1^{05} based on Zermelo's 1908 argument Z_{29}^{08} for the Axiom of Choice:

$$I_1 = (\mathsf{Assume}_\neg(\text{existence\_proof\_requires\_definition}))$$
$$I_2 = (I_1 \rightsquigarrow \neg\text{accept}(AC))$$
$$I_3 = (Z_{29}^{08}, I_2 \rightsquigarrow \bot)$$
$$I_4 = (\mathsf{ProofbyContrad}(I_3, \neg\text{existence\_proof\_requires\_definition}))$$

The idea is that assuming a premise ("existence_proof_requires_definition") of Lebesgue's argument against the Axiom of Choice, we can derive that the Axiom of Choice should not be accepted, which in combination with Zermelo's argument for the acceptance of the Axiom of Choice leads to a contradiction. So we have a proof by contradiction for ¬existence_proof_requires_definition, which thus attacks Lebesgue's argument. The relevance of this argument to the formal properties of our model is explained in Section 1.7 of the technical online appendix [81].

While the model described here has not led to philosophically relevant implicit arguments, we believe that the methodology we are proposing has the potential to bring to light such arguments once more sophisticated formal models of debates in the formal sciences are constructed. We expect the use of automated theorem provers to be helpful in order to discover philosophically relevant implicit arguments in more sophisticated models, just like they already have been used by [69] and [70] to discover philosophically relevant mistakes and insights in axiomatic theories of metaphysics, as explained in the last paragraph of the Introduction. This would allow for the discovery of mistakes and new insights at the meta-level of debates about formal theories rather than just at the object level of the theories themselves.

We consider it one of the strengths of our methodological approach that it allows to identify such implicit arguments that no one has put forward, but that could be put forward and that could have a relevant influence on the outcome of the debate.

Without imposing preferences on the set of rules, all attacks in our model other than the just mentioned undercuts would become *practically* bidirectional. By this we mean that even though there can be a unidirectional attack from some argument A to some argument B, in such a case there will always be an attack back onto A from some argument B' that is closely related to B and accepted in the same circumstances as B. In order to make the model more interesting and more realistic, we have therefore include in it a preference order on the rules, which by Definition 6.3.4 gives rise to a preference order on the arguments. One drawback of our methodology is that it gives no methodological guidance on how to select a preference order on the rules, which is the main determining factor for which extensions are finally accepted. In our model, we followed our common sense of the relative strength of different arguments from the historical debate in order to specify the preference order between the rules.

The set of rules of our model allow for infinitely many arguments to be constructed, so that the EAF corresponding to the model will also be infinite. However, only a small finite subset of this infinite EAF contains attacks that are relevant for the overall status of the acceptability of the Axiom of Choice, which was the focus of attention of the debate that we have formally modeled. In Figure 6.3, we depict the small subset of relevant arguments and the defeats between them. In this depiction, the letter in the argument name (Z, P, L or H) refers to either Zermelo, Peano, Lebesgue or Hadamard as the source of the argument, and the subscript indicates the year in which the argument was presented (with the 19 dropped, as they were all presented between 1904 and 1908). For the precise content of the argument and the details of their formalization in ASPIC-END, please refer to the technical online appendix [81]. Here we concisely sketch the content of the arguments that have not yet been specified above:

- P_2^{06}: Peano points out that in an 1890 publication he had already considered and rejected the assumption that infinitely many arbitrary choices can be made in an argument.

- P_{14}^{06}: Peano points out that while a single arbitrary choice and thus any finite number of arbitrary choices can be formalized in his *Formulario Mathematico*, an infinite number of arbitrary choices would require an infinitely long argument, which is not allowed in his *Formulario Mathematico*. This argument has the implicit premise that an argument can be accepted if and only if it can be formalized in the *Formulario Mathematico*.

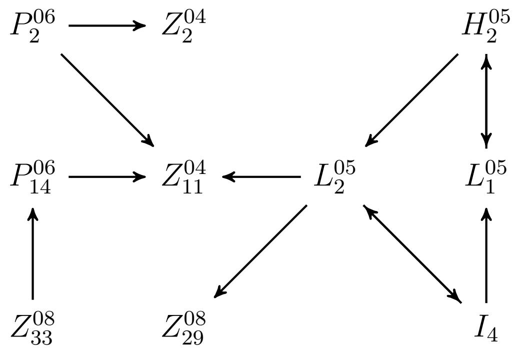


Figure 6.3: The relevant arguments and attacks from the example

- Z_{29}^{08}: Zermelo points out that Peano himself arrived at the fundamental principles of his *Formulario Mathematico* by analyzing the rules of inference that have historically been recognized as valid and by referring both to the intuitive evidence for the rules and to their necessity for science. He then argues that the Axiom of Choice can be justified in the same way: Multiple set theorists have implicitly applied it, which supports both the claim that it has historically been recognized as valid and that it is intuitively evident. Furthermore, Zermelo lists seven theorems which he believed not to be provable without the Axiom of Choice, and concludes that the Axiom of Choice is necessary for science.

- Z_{33}^{08}: The implicit assumption in P_{14}^{06} (see above) is incorrect, because by Z_{29}^{08} arguments using the Axiom of Choice can be accepted even though they cannot be formalized in Peano's *Formulario Mathematico*.

- H_2^{05}: Hadamard argues against Lebesgue's premise that an existence proof requires definition by pointing out that historical progress in mathematics was achieved by annexing notions which had previously been considered to be outside mathematics because it was impossible to describe them.

Restricted to this set of relevant arguments, there are two argumentative core (AC) extensions: $S_1 = \{P_2^{06}, Z_{33}^{08}, Z_{29}^{08}, H_2^{05}, I_4\}$, and $S_2 = \{P_2^{06}, Z_{33}^{08}, L_1^{05},\ L_2^{05}\}$. This means that arguments P_2^{06} and Z_{33}^{08} are accepted in every AC-extension of our model, while P_{14}^{06}, Z_2^{04} and Z_{11}^{04} are rejected in every AC-extension, and the status of the arguments Z_{29}^{08}, I_4, L_1^{05} and L_2^{05} depends on the choice of AC-extension. This set of relevant arguments contains two arguments with conclusion accept(AC), namely Z_{11}^{04} and Z_{29}^{08}. While the first one gets rejected in both extensions, the second one gets accepted in one and rejected in the other extension, so that overall, the status of the claim accept(AC) depends on the choice of the AC-extension.

These properties of our formal model intuitively correspond to the situation that on the one hand there are compelling arguments both in favor and against the Axiom of Choice, and purely formal methods will not decide which of the two stands is "correct" (if there even is a single "correct" answer here), while on the other hand certain arguments in favor or against the Axiom of Choice are so weak that they do not hold up against the scrutiny provided by certain counterarguments against them.

Of course, the fact that the status of the Axiom of Choice in our formal model of the debate is not determined but depends on the choice of the AC-extension is to a certain extent an artifact of the choice of arguments that we formalized and of the preference order that we imposed. We could have gotten a different result, for example if we had chosen to formalize only strong arguments in favor of the Axiom of Choice and weak arguments against it, or if we had just made significantly different judgments about the preference order on the rules involved in our model. So at the current level of development, such a model cannot be seriously defended as a method for deciding which side in a debate is right. What it can do, however, is to help us discover relevant implicit arguments like argument I_4 in our model (and hopefully with a more developed model also philosophically more relevant implicit arguments), to help us get a more precise understanding of what assumptions are made and what is at stake in a given debate, and to point towards weaknesses of the current methodology of structured argumentation theory, like the lack of a methodological guidance for choosing a preference order on the rules.

6.6 Closure and rationality postulates

In this section, we present four rationality postulates that ASPIC-END satisfies and that are analogous to the four postulates that [66] have established for ASPIC+, as well as two new postulates motivated by the application of structured argumentation to debates in the formal sciences.

The first postulate concerns the closure of the extensions under the sub-argument relation. The idea is that one cannot accept an argument while rejecting part of it.

Theorem 6.6.1. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ and S be an AC-extension of F. Then, for all $A \in S$, $\mathsf{Sub}(A) \subseteq S$.*

The proof of Theorem 6.6.1 rests on the following lemma, which can be proven in a straightforward way as in the case of ASPIC+ (see Lemma 35 of [66]):

Lemma 6.6.2. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ, $S \subseteq \mathcal{A}$ and $A, B \in \mathcal{A}$. We have that:*

1. *If S defends A and $S \subseteq S'$, then S' defends A.*
2. *If A defeats B' and $B' \in \mathsf{Sub}(B)$, then A defeats B.*
3. *If S defends A and $A' \in \mathsf{Sub}(A)$, then S defends A'.*

We now show another intuitive result which will be needed in the proof of the postulates. This result is that given a satisfactory set of arguments, including additional arguments which do not interfere with the admissibility of the set, does not prevent the set from being satisfactory.

Lemma 6.6.3. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ and $S, S' \subseteq \mathcal{A}$ with S satisfactory. If S' is admissible and $S \subseteq S'$, then S' is also satisfactory.*

Proof:

Assume $S' \supseteq S$ is admissible. Now, suppose for a contradiction that S' is not satisfactory. Then, since S' is admissible, there exists $S'' \supset S'$ such that $S'' >_p S'$ and S'' is admissible.

We will now show that $S'' >_p S$. Since $S'' \supset S'$ and $S' \supseteq S$, we have $S'' \supset S$. For each explanandum e for which S offers an explanation $X[e]$, $X[e] \in S''$, so S'' also offers an explanation for e. Hence, S'' offers an explanation for at least as many explananda as S. However, since $S'' >_p S'$, there exists an explanandum e' for which S'' offers an explanation but for which S' does not offer an explanation. $S \subseteq S'$, hence S does not offer an explanation for e' either. Therefore, S'' offers an explanation for strictly more explananda as S and thus $S'' >_p S$.

So we have $S'' \supset S$, $S'' >_p S$ and S'' is admissible. However, since S is satisfactory, this is a contradiction. Hence, S' is satisfactory. □

Proof of Theorem 6.6.1:

Let $A \in S$ and $A' \in \mathcal{A}$. Assume $A' \in \mathsf{Sub}(A)$. Suppose for a contradiction that $S \cup \{A'\}$ is not conflict-free. Since S is an AC-extension of F, S is conflict-free. Hence, either A' defeats some argument $B \in S$, or some argument $B \in S$ defeats A'.

- Suppose first that A' defeats some argument $B \in S$. Then, since S is an AC-extension of F, there exists some argument $B' \in S$ which defeats A'. Thus, by Lemma 6.6.2.2, B' also defeats A. But S is conflict-free. We have a contradiction.

- Suppose now that some argument $B \in S$ defeats A'. Then, by Lemma 6.6.2.2, B also defeats A. But S is conflict-free. We have a contradiction.

Since both cases lead to a contradiction, we can conclude that $S \cup \{A'\}$ is conflict-free.

Now, S defends A and so, by Lemma 6.6.2.3, S defends A'. Since S is an AC-extension of F, S also defends S. Thus, S defends $S \cup \{A'\}$. Hence, by Lemma 6.6.2.1, $S \cup \{A'\}$ defends $S \cup \{A'\}$. Since $S \cup \{A'\}$ is also conflict-free, $S \cup \{A'\}$ is admissible.

Also, by Lemma 6.6.3, since S is satisfactory and $S \cup \{A'\}$ is admissible, $S \cup \{A'\}$ is also satisfactory.

Now suppose for a contradiction that $A' \notin S$. Then, $S \cup \{A'\}$ is a proper superset of S which is also satisfactory. Hence, S is not an AC-extension of F. So we have a contradiction, and thus $A' \in S$. □

Notice that this postulate does not hold for EC-extensions, as they are by definition minimal in their inclusion of arguments, and thus will often leave out low-level sub-arguments.

The second postulate concerns the closure of the conclusions under intuitively strict rules. In the case of ASPIC+, the corresponding postulate concerned the closure of the conclusions under all strict rules (see Theorem 13 in [66]). But since ASPIC-END allows for the rejection of intuitively strict rules, it is undesirable to consider the closure under all of them. Instead, we consider the closure under a set of intuitively strict rules which are deemed acceptable. The following two definitions define the set of *accepted* intuitively strict rules and the *closure* under a given set of intuitively strict rules:

Definition 6.6.1 (Set of Accepted Intuititvely Strict Rules)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow,$
$\dashrightarrow\rangle$ be the EAF defined by Σ and S be an extension of F. The *set of intuitively strict rules accepted by* S is $\mathcal{R}_{isa}(S) = \{r \in \mathcal{R}_{is} | \forall A \in \mathcal{A}$ s.t. $\mathsf{As}(A) = \emptyset$ and $\mathsf{Conc}(A) = \neg n(r)$ or $\neg\mathsf{Conc}(A) = n(r), \exists B \in S$ s.t. B defeats $A\}$.

Definition 6.6.2 (Closure of a Language under a Set of Rules)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $P \subseteq \mathcal{L}$ and $R' \subseteq \mathcal{R}_{is}$. We define the *closure of* P *under the set of rules* R', denoted $Cl_{R'}(P)$, as the smallest set such that $P \subseteq Cl_{R'}(P)$, and when $(\varphi_1, ..., \varphi_n \rightsquigarrow \psi) \in R'$ and $\varphi_1, ..., \varphi_n \in Cl_{R'}(P)$, then $\psi \in Cl_{R'}(P)$.

Now the postulate on the closure under accepted intuitively strict rules can be formulated as follows:

Theorem 6.6.4. *Let* $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ *be an argumentation theory,* $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow,$
$\dashrightarrow\rangle$ *be the EAF defined by* Σ *and* S *be an AC-extension of* F*. Then,* $\mathsf{Conc}(S) = Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$*.*

Proof:

Let S be an AC-extension of F. We want to show that $\mathsf{Concs}(S) =$
$Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$. First, notice that $\mathsf{Concs}(S) \subseteq Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$. Hence, we only need to show that if $(\varphi_1, ..., \varphi_n \rightsquigarrow \psi) \in \mathcal{R}_{isa}(S)$ and $\varphi_1, ..., \varphi_n \in \mathsf{Concs}(S)$, then $\psi \in \mathsf{Concs}(S)$.

Suppose that $(\varphi_1, ..., \varphi_n \rightsquigarrow \psi) \in \mathcal{R}_{isa}(S)$ and $\varphi_1, ..., \varphi_n \in \mathsf{Concs}(S)$. Then, by the definition of Concs, there exists $A_1, ..., A_n \in S$ such that $\mathsf{Conc}(A_i) = \varphi_i$ and $\mathsf{As}(A_i) = \emptyset$ for $1 \leq i \leq n$. Hence, we can construct the argument $A = A_1, ..., A_n \rightsquigarrow \psi$, and thus $A \in \mathcal{A}$ with $\mathsf{As}(A) = \emptyset$.

Assume $B \in \mathcal{A}$ defeats A. Then, B either undercuts, assumption-attacks or successfully rebuts A. Let us first consider the case of undercut. Then, $\mathsf{As}(B) = \emptyset$ and either $\mathsf{Conc}(B) = \neg n(\varphi_1, ..., \varphi_n \rightsquigarrow \psi)$ or $\neg\mathsf{Conc}(B) = n(\varphi_1, ..., \varphi_n \rightsquigarrow \psi)$. However, since $(\varphi_1, ..., \varphi_n \rightsquigarrow \psi) \in \mathcal{R}_{isa}(S)$, there exists $C \in S$ such that C defeats B. Also, since A cannot be rebutted nor assumption-attacked, S defends A.

Now suppose there is an argument $D \in S$ such that D defeats A. Then, since S defends A, there is an argument $C \in S$ which defeats D. However, S is conflict-free, so we have a contradiction. Hence, there is no argument in S which defeats A.

Let us now assume A defeats some argument $D \in S$. Since S is admissible, there is an argument in S which defeats A. However, we have just concluded that no such argument exists, hence we have a contradiction. Therefore, A does not defeat any of the arguments in S.

Thus, $S \cup \{A\}$ is conflict-free. Also, since S defends A, $S \cup \{A\}$ is admissible. By Lemma 6.6.3 and since S is satisfactory, $S \cup \{A\}$ is also satisfactory.

Assume $A \notin S$. Then, since $(S \cup \{A\}) \supset S$ is satisfactory, S is not an AC-extension of F. This is a contradiction of one of our initial assumptions. Hence, $A \in S$. Therefore, $\psi \in \mathsf{Concs}(S)$ and thus $\mathsf{Concs}(S) = Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$. □

The last two postulates presented by [66] are direct and indirect consistency, which state that when the set of strict rules is consistent, the set of conclusions and the closure of this set under strict rules are consistent.

While consistency postulates are not relevant for the application of ASPIC-END to argumentation about paradoxes, we also want ASPIC-END to be applicable to more standard domains in which the consistency postulates are relevant. For this reason, we also establish consistency postulates for ASPIC-END.

In order to show the consistency of the conclusions, we will have to show that no two arguments with contradictory conclusions may co-exist in the same extension. While these two arguments may have intuitively strict TopRules, and thus not attack each other, we will show that one of their sub-arguments is attacked and undefended. For the purpose of gradual inspection of the sub-arguments, we first define direct sub-arguments.

Definition 6.6.3 (Direct Sub-Argument)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ the EAF defined by Σ and $A, A' \in \mathcal{A}$. We say that A' is a *direct sub-argument* of A iff $A' \in \mathsf{Sub}(A)$ and there is no $A'' \in \mathsf{Sub}(A)$ s.t. $\mathsf{Sub}(A') \subset \mathsf{Sub}(A'')$.

Then, in order to identify those potential points of attack, we define maximal fallible sub-arguments, which represent the top-most sub-arguments with defeasible top rules.

Definition 6.6.4 (Maximal Fallible Sub-Arguments)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ the EAF defined by Σ and $A \in \mathcal{A}$. We define the multiset $M(A)$ of the *maximal fallible sub-arguments* of A as:

$$M(A) := \begin{cases} \{A\} & \text{if } \mathsf{TopRule}(A) \in \mathcal{R}_d \\ \emptyset & \text{if } \mathsf{DefRules}(A) = \emptyset \\ \biguplus_{i=1}^{k} M(A_i) & \text{Otherwise, where } \{A_1, ..., A_k\} \text{ is} \\ & \text{the set of direct sub-arguments of A.} \end{cases}$$

For a set of arguments S, we write $\mathsf{Subs}(S)$ as a shorthand for $\bigcup_{A \in S} \mathsf{Sub}(A)$.

Definition 6.6.5 (Intuitively Strict Continuation)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ the EAF defined by Σ and $S \subseteq \mathcal{A}$. We say that $A \in \mathcal{A}$ is an *intuitively strict continuation* of S iff:

- $\mathsf{Subs}(S) \subseteq \mathsf{Sub}(A)$;
- $\{r \mid \text{for some } X \in \mathsf{Sub}(A) \setminus \mathsf{Subs}(S), r = \mathsf{TopRule}(X)\} \subseteq R_{is}$

We then show some intuitive results from our preference lifting. These results are closely related to the properties of a reasonable argument ordering as defined in [66].

Lemma 6.6.5. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory and $\prec$ the preference relation over $\mathcal{A}$ lifted from $<$. We have that:*

1. *for all $A, B \in \mathcal{A}$, if $\mathsf{DefRules}(A) = \emptyset$, then $A \not\prec B$;*
2. *for all $A, B \in \mathcal{A}$, if $\mathsf{DefRules}(A) = \emptyset$ and $\mathsf{DefRules}(B) \neq \emptyset$, then $B \prec A$;*

3. *for any finite multiset $\{C_1, ..., C_n\}$ of arguments, it is not the case that for all $i \in \{1, ..., n\}$, $C^{+\setminus i} \prec C_i$ (where $C^{+\setminus i}$ is an intuitively strict continuation of $\{C_1, ..., C_n\} \setminus \{C_i\}$).*

Proof:

1. Suppose for a contradiction that $A \prec B$. Then, by definition, there exists $r_a \in \textsf{DefRules}(A)$ such that for all $r_b \in \textsf{DefRules}(B)$, $r_a < r_b$. However, since $\textsf{DefRules}(A) = \emptyset$, no such r_a exists. Hence, $A \not\prec B$. □
2. Take any $r_b \in \textsf{DefRules}(B)$. Since $\textsf{DefRules}(A) = \emptyset$, it holds that for all $r_a \in \textsf{DefRules}(A)$, $r_b < r_a$. Hence, $B \prec A$. □
3. Suppose for a contradiction that for all $i \in \{1, ..., n\}$, there exists an intuitively strict continuation $C^{+\setminus i}$ such that $C^{+\setminus i} \prec C_i$. Take an arbitrary C_j with $1 \leq j \leq n$. We have that there exists $C^{+\setminus j}$ such that $C^{+\setminus j} \prec C_j$. Hence, there exists $r \in \textsf{DefRules}(C^{+\setminus j})$ such that for all $r_j \in C_j$, $r < r_j$. Select a least preferred such r (for all $r_l \in \textsf{DefRules}(C^{+\setminus j})$, $r_l \not< r$). Take any argument $C_k \in \{C_1, ..., C_n\}$ such that $r \in \textsf{DefRules}(C_k)$. Since $r < r_j$ for all $r_j \in C_j$ and $r_l \not< r$ for all $r_l \in \textsf{DefRules}(C^{+\setminus j}) = \bigcup_{i=1, i \neq j}^{i=n} \textsf{DefRules}(C_i)$, we have that $r_m \not< r$ for all $r_m \in \bigcup_{i=1}^{i=n} \textsf{DefRules}(C_i)$, and hence $r_m \not< r$ for all $r_m \in \bigcup_{i=1, i \neq k}^{i=n} \textsf{DefRules}(C_i)$. For all intuitively strict continuations $C^{+\setminus k}$ of $\{C_1, ..., C_{k-1}, C_{k+1}, ..., C_n\}$, we have $\textsf{DefRules}(C^{+\setminus k}) = \bigcup_{i=1, i \neq k}^{i=n} \textsf{DefRules}(C_i)$. Hence, we have $C^{+\setminus k} \not\prec C_k$. This is a contradiction, and hence it is not the case that for all $i \in \{1, ..., n\}$, $C^{+\setminus i} \prec C_i$. □

We have three requirements for applying the consistency postulates. The first is that there cannot be non-defeasible arguments which contradict each other. The second requirement ensures that a formula and its negation are considered as contradictory and the third guarantees that no assumptions for proof by contradiction are prevented. The last two requirements are motivated by the consideration that in the applications of ASPIC-END not related to paradoxes, one would likely accept classical or intuitionistic logic, for both of which these requirements hold.

Definition 6.6.6 (Consistency-Inducing)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory. We say that Σ is *consistency-inducing* iff:

1. there are no $A, B \in \mathcal{A}$ such that $\textsf{DefRules}(A) = \textsf{DefRules}(B) = \emptyset = \textsf{As}(A) = \textsf{As}(B)$ and $\textsf{Conc}(A) = \neg\textsf{Conc}(B)$,
2. for each $\varphi \in \mathcal{L}$ there is a rule r_φ of the form $\varphi, \neg\varphi \rightsquigarrow \bot \in \mathcal{R}_{is}$ such that $n(r_\varphi)$ is undefined,
3. there is no rule $r \in \mathcal{R}$ such that $Assumable_\neg(\varphi)$ appears in r.

The following theorem establishes direct consistency for ASPIC-END:

Theorem 6.6.6. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be a consistency-inducing argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ and S be an AC or EC-extension of F. Then, there does not exist $\varphi \in \mathsf{Concs}(S)$ such that $\neg\varphi \in \mathsf{Concs}(S)$.*

Proof. Suppose for a contradiction that there exists $\varphi \in \mathsf{Conc}(S)$ such that $\neg\varphi \in \mathsf{Conc}(S)$. Then, there exist two arguments $A, B \in S$ such that $\mathsf{Conc}(A) = \varphi$, $\mathsf{Conc}(B) = \neg\varphi$ and $\mathsf{As}(A) = \emptyset = \mathsf{As}(B)$.

Consider the multiset $S = M(A) \uplus M(B)$. By Lemma 6.6.5.3, there exists $C \in S$ such that for all intuitively strict continuations S' of $S \setminus \{C\}$, we have $S' \nless C$. Without loss of generality, suppose $C \in M(A)$. Let $C' = Assume_{\neg}(\mathsf{Conc}(C))$ and construct A' from A by replacing C with C'. We now have that $\mathsf{Conc}(A') = \varphi$ and $\mathsf{As}(A') = \{\mathsf{Conc}(C)\}$. Since Σ is consistency-inducing, $\varphi, \neg\varphi \rightsquigarrow \bot \in \mathcal{R}_{is}$. Thus, we can construct $A'' = A', B \rightsquigarrow \bot$ with $\mathsf{As}(A'') = \{\mathsf{Conc}(C)\}$. Hence, we can also construct $D = \mathsf{ProofByContrad}(\neg\mathsf{Conc}(C), A'')$. Since $\mathsf{Conc}(D) = \neg\mathsf{Conc}(C)$, D attacks C. Also, D is an intuitively strict continuation of $S \setminus C$, thus we have $D \nless C$ and therefore D defeats C.

By Theorem 6.6.1, since $C \in \mathsf{Sub}(A)$, $C \in S$.

Similarly, for all $A_i \in M(A)$, $A_i \in S$ by Theorem 6.6.1. A' is an intuitively strict continuation of $M(A) \setminus C$ which uses the same rules as A. Hence, S defends A', and thus $A' \in S$.

Suppose an argument F defeats D. Then, F cannot defeat D on A'' by rebut since $\mathsf{TopRule}(A'') \in \mathcal{R}_{is}$. Also, F cannot defeat D on A'' by undercutting, since Σ is consistency-inducing and thus $n(\mathsf{TopRule}(A''))$ is undefined. F cannot defeat D nor A'' on C' by $\neg$-assumption-attack, again because Σ is consistency-inducing. Since $D =$ $\mathsf{ProofByContrad}(\neg\mathsf{Conc}(C), A'')$, F cannot defeat D on D either.

So F defeats D on D', where $D' \neq A''$, $D' \neq D$ and $D \neq C'$. Hence, $D' \in \mathsf{Sub}(A')$ or $D' \in \mathsf{Sub}(B)$. By Theorem 6.6.1 and since $A', B \in S$, we have $D' \in S$. Hence, S defends D from F and so $D \in S$.

But D defeats C, so S is not conflict-free, which is a contradiction. Therefore, no such $\varphi \in \mathsf{Concs}(S)$ exists and thus $\mathsf{Concs}(S)$ is consistent.

□

Indirect consistency of AC-extensions follows from closure under accepted intuitively strict rules together with direct consistency:

Theorem 6.6.7. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be a consistency-inducing argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ and S be an AC-extension of F. Then, there does not exist $\varphi \in Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$ such that $\neg\varphi \in Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$.*

We want ASPIC-END to be applicable to debates in the formal sciences, in which the correctness of logical rules can be up for debate. For example, among the proposals made by philosophers of how to handle the semantic paradoxes, there is paraconsistent dialetheism [86], which accepts some inconsistencies as true and uses a paraconsistent logic to avoid that everything can be derived. And in order to be able to show the internal structure of the paradox, we need to have an inconsistency arise from intuitively strict rules under no assumptions. For these reasons, the consistency postulates do not make sense for this kind of application of ASPIC-END.

However, there is a property similar to consistency that should still hold even when the intuitively strict rules lead to paradoxes and when the output extensions contain one that accepts paraconsistent dialetheism, namely that an extension should never be trivial, i.e. conclude everything.

For the non-triviality of the extensions, we require every intuitively strict rule, except for the ones of conjunction elimination from $\bot$, to have a name so that it can be attacked. We say that the argumentation theory is well-defined if it satisfies this requirement, and assume well-definedness in the non-triviality postulate stated in Theorem 6.6.8.

Definition 6.6.7 (Well-Defined Argumentation Theory)**.** Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be an argumentation theory. We say that Σ is *well-defined* if and only if for each rule $r' \in \mathcal{R}_{is} \setminus \mathcal{R}_{ce}$, $n(r') \in \mathcal{L}$.

Theorem 6.6.8. *Let $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ be a well-defined argumentation theory, $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ be the EAF defined by Σ, and S be an AC or EC-extension of F. Then, $\bot \notin \mathsf{Concs}(S)$.*

Proof. Suppose for a contradiction that $\bot \in \mathsf{Concs}(S)$. Then there exists a minimal (under sub-argument relation) argument $A \in S$ such that $\mathsf{Conc}(A) = \bot$ and $\mathsf{As}(A) = \emptyset$.

We have two cases:

1. $\mathsf{TopRule}(A)$ is undefined. Then, A must be of the form $\mathsf{ReasonByCases}(\bot, A_1, A_2, A_3)$. We have two sub-cases:
   (a) $\mathsf{DefRules}(A) \neq \emptyset$. Then, there must be a minimal (w.r.t. $\prec$) argument A' such that $\mathsf{TopRule}(A') \in \mathcal{R}_d$. Let $B = A \rightsquigarrow \neg\mathsf{Conc}(A')$. Then, since A' is minimal w.r.t. $\prec$, $B \not\prec A'$ and so B successfully rebuts A', so B defeats A.
   (b) $\mathsf{DefRules}(A) = \emptyset$. If A_3 is of the form $\mathsf{ReasonByCases}(\bot, A'_1, A'_2, A'_3)$, set $A_3 := A'_3$ and repeat this process until you obtain an argument A_3 which is not a reasoning by cases. Now A_3 is such that $\mathsf{Conc}(A_3) = \phi \vee \neg\phi$ for some $\phi \in \mathcal{L}$ and $\mathsf{DefRules}(A_3) = \emptyset$ since $\mathsf{DefRules}(A) = \emptyset$, so A_3 must be of the form $P_1, P_2, ..., P_n \rightsquigarrow \phi \vee \phi'$. Since Σ is well-defined, $\mathsf{TopRule}(A_3)$ is defined, and so let $B = A \rightsquigarrow \neg n(\mathsf{TopRule}(A_3))$. So B undercuts A_3 and thus defeats A.

2. $\mathsf{TopRule}(A)$ is defined. Let $r = \mathsf{TopRule}(A)$. If $r \in \mathcal{R}_{is}$, then $n(r) \in \mathcal{L}$ and so let $B = A \rightsquigarrow \neg n(r)$. Otherwise, let $B = A \rightsquigarrow \neg\bot$. By the definition of $\prec$ and the construction of B, $B \not\prec A$. Then B undercuts or successfully rebuts A on A, so B defeats A.

Since S is an AC- or EC-extension of F, it defends itself, so there exists $C \in S$ such that C defeats B. Suppose for a contradiction that C defeats B on $B' \neq B$. Since $\mathsf{Sub}(B) = \mathsf{Sub}(A) \cup \{B\}$, $B' \in \mathsf{Sub}(A)$. Then, by Lemma 6.6.2.2, C defeats A on B'. But S is conflict-free, so we have a contradiction. Hence, C defeats B on B. Since $B = A \rightsquigarrow \neg n(r)$, B cannot be rebutted nor assumption-attacked. Hence, C undercuts B on B. But since $\mathsf{TopRule}(B) \in \mathcal{R}_{ce}$, $n(\mathsf{TopRule}(B))$ is undefined, i.e. no argument undercuts B on B, a contradiction.

Hence, $\bot \notin \mathsf{Concs}(S)$.

□

Indirect non-triviality of AC-extensions then follows from closure under accepted intuitively strict rules and direct non-triviality:

Theorem 6.6.9. *Let* $\Sigma = (\mathcal{L}, \mathcal{R}, n, <)$ *be a well-defined argumentation theory,* $F = \langle \mathcal{A}, \mathcal{X}, \rightarrow, \dashrightarrow \rangle$ *be the EAF defined by* Σ *and* S *be an AC-extension of* F*. Then,* $\bot \notin Cl_{\mathcal{R}_{isa}(S)}(\mathsf{Concs}(S))$.

One problem that ASPIC-END shares with ASPIC+ and that we have left for future work is that it does not satisfy the non-interference postulate, which says that in a theory consisting of two syntactically separate parts, the outcome of each part must be independent of the other part [87]. We would like to briefly discuss this postulate in order to show that it does not pose any problems to the example applications of ASPIC-END that are presented in the chapter and in this technical appendix.

A common example that illustrates why ASPIC+ does not satisfy this postulate is base on an innocent bystander, i.e. a defeasible premise $\Rightarrow r$ with the property that r is not used in the rest of the theory, and on strict rules that allow explosive conclusions from a contradiction (*ex falso quodlibet*) – e.g. strict rules that correspond to inferences in classical logic. If there is any conflict in the theory arising from defeasible reasoning, then the rules of classical logic allow to conclude $\neg r$, thus producing an argument that attacks the argument for r. This phenomenon is called *contamination* in the literature. If every extension decides the conflict on which this argument is based one way or the other, then this argument attacking r will be rejected in every extension. If on the other hand the conflicting arguments are themselves attacked by unresolved self-attacking arguments (or by some other paradoxical arguments, i.e. arguments involved in an unresolved odd cycle), then the argument attacking r and the argument for r will both have an undecided status, so that the argument for r is affected by the theory that does not mention r. In formalizations of this example in the literature, Liar-like sentences have been used in the place of the self-attacking arguments needed for this construction (cf. Example 6.2 in [88]). For this reason, one could be worried that our application of ASPIC-END to modeling the Liar paradox suffers from this problem. We will now briefly explain why this does not cause problems for our example application about the Liar paradox, and also why our ASPIC-END model of the debate about the Axiom of Choice does not suffer from this problem.

First we would like to point out that there are crucial differences between ASPIC-END formalizations of semantic paradoxes like the Liar sentence on the one hand and the contamination examples considered in the literature on the other hand. A semantic paradox formalized in ASPIC-END gives rise to an intuitively strict argument that has conclusion $\bot$. We consider debates about how to resolve such arguments, i.e. about different possibilities to undercut an intuitively strict rule in such an argument. Once such an argument is successfully undercut, it no longer gives rise to contamination issues. On the other hand, the standard contamination examples consist of two defeasible arguments that have contrary conclusions and that are not attacked by any set of arguments that can be accepted together.

Importantly, contamination does not affect our example ASPIC-END models in a substantial way. The ASPIC-END model of debate on the Liar paradox does not contain any explosive rule of the form $\phi, \neg\phi \rightsquigarrow \bot$, so as it stands it certainly does not suffer from contamination. But even if one were to add an innocent bystander $\Rightarrow r$ and the explosive rule $\mathit{accept}(\mathit{classical\_logic}), \neg\mathit{accept}(\mathit{classical\_logic}) \rightsquigarrow \neg r$, it would still be the case that r is accepted in both AC-extensions, as one of them has a justification (namely argument B_2) for rejecting $\neg\mathit{accept}(\mathit{classical\_logic}$ and the other one has a justification (namely argument C_2) for rejecting $\mathit{accept}(\mathit{classical\_logic}$. The only way to add contamination to the example would be by making both argument B_2 and argument C_2 being attacked by an ambiguous argument, e.g. an argument that is in an odd cycle that does not get attacked from the outside. The ASPIC-END model of the debate about the Axiom of Choice does contain explosive rules that cannot be undercut, so given an innocent bystander $\Rightarrow r$, we could for example construct the argument $(L_1^{05}, H_2^{05} \rightsquigarrow \bot) \rightsquigarrow r$, but again r would be accepted in both AC-extensions as one of them rejects L_1^{05} and the other one rejects H_2^{05}. Again, contamination is avoided due to the fact that is not possible to construct unresolvable odd cycles that attack the two arguments L_1^{05} and H_2^{05}.

6.7 Conclusion and Future Work

We have proposed the application of the structured argumentation methodology to formally model informal and semi-formal debates in the formal sciences. For this purpose, we have proposed a modification of ASPIC+ called ASPIC-END, which incorporates a formal model of explanations, and features natural-deduction style arguments. We have then discussed two instantiations of ASPIC-END, one that models relatively simple arguments about two solutions the Liar Paradox, and one that constitutes a more extensive model of part of the debate that mathematicians had about the Axiom of Choice in the early 20th century.

In a technical online appendix [81] we have proved four rationality postulates for ASPIC-END that are analogous to the four postulates that [66] have established for ASPIC+, as well as two new postulates motivated by the application of structured argumentation to debates in the formal sciences. One problem that ASPIC-END shares with ASPIC+ and that we have left for future work is that it does not satisfy the non-interference postulate [87].

As explained in the introduction, we believe the methodological approach proposed in this chapter to be of significant potential for further research. The model of the debate about the Axiom of Choice sketched in Section 6.5 could be extended to a model covering a wider range of topics related to the foundational questions in mathematics as well as active research questions in philosophical logic. Given that with increasing size of the model it becomes more and more difficult to produce the model manually and to find all relevant arguments and attacks, we propose that interactive theorem provers like Isabelle [89] or HOL Light [90] be used for producing and studying such extensive formal models.

Furthermore, combining the methodology of structured argumentation theory with insights from natural language semantics could lead to formal models that are more faithful to the logical form implicit in natural language, which could strengthen the link between the formalization of a debate and the original natural language form of the debate.

Chapter 7

Future work

In this chapter we present some preliminary results on topics related to the work described in the earlier chapters. We first investigate in Sec. 7.1 the connections between commitment graphs as described in Chapter 3, and multi-agent argumentation and dialogues.

The work on flattening from Chapter 5 also raises the question of which relations can be flattened into a classical argumentation framework. We present some preliminary answers to this question in Sec. 7.2.

7.1 Multi-agent dialogues

In this section, we present some preliminary work on the dialogical and dynamical aspects in multi-agent argumentation. Based on the work of laid out in Chapter 3 which refines extensions into commitment graphs and the work on multi-agent argumentation started by Arisaka et al. [15], we observe the process of argument-sharing between the agents and how the individual attitude of each agent affect the global outcome of the argumentation process.

We first provide a few preliminary definitions from the work of Arisaka et al. [15] and then present the multi-agent dialogue framework.

7.1.1 Conditional and multi-agent argumentation

A multi-agent argumentation framework is an argumentation framework together with a set of agents and an assignment of the arguments to the agents. We call the agents also the *sources* of the arguments.

Definition 7.1.1 (Multi-agent argumentation)**.** A multi-agent argumentation framework is a tuple $\langle \mathcal{A}, \rightarrow, Ag, Src \rangle$ where $\langle \mathcal{A}, \rightarrow \rangle$ is an argumentation framework, Ag is a set called agents and $Src : \mathcal{A} \rightarrow Ag$ is a function mapping each argument to the agent that put it forward (also known as its source).

For the semantics, we first define individual acceptance by an agent. We consider the part of the multi-agent framework that is relevant to the agent, which we call the *agent argumentation framework*. It contains its own arguments together with the attacks among them, the relevant arguments of other agents, an extension of these other arguments, and an attack relation from the other arguments to its own arguments. The agent semantics considers the agent argumentation framework, as well as the arguments accepted by other agents. This conditional acceptance is called a local function by Baroni *et al.* [57] call a local function.

We slightly rewrite the definition of local function to make it explicit that acceptance of arguments by agents is conditional on the acceptance of arguments by other agents. Moreover, in contrast to Baroni et all, we do not consider the attacks among input arguments. Since Baroni *et al.* define their local acceptance functions for all Dung semantics, not only for stable semantics, their definitions are more general than ours. Similar notions are defined also by Liao [91]. We refer to these papers for further explanations and examples of local functions.

Definition 7.1.2 (Individual conditional acceptance)**.** For multi-agent argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src \rangle$, the *argumentation framework of agent* A is a tuple $\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle$, where $\mathcal{A}_A = \{a \in \mathcal{A} \mid Src(a) = A\}$ are the arguments of agent A, $\rightarrow_A = \rightarrow \cap (\mathcal{A}_A \times \mathcal{A}_A))$ are its attacks, $I_A = \{a \in \mathcal{A} \mid a \notin \mathcal{A}_A, (a,b) \in \rightarrow, b \in \mathcal{A}_A\}$ are the relevant arguments from other agents, and $R_{I_A} = \rightarrow \cap (I_A \times \mathcal{A}_A)$ is the corresponding attack relation. The stable semantics of agent A and context $E_{I_A} \subseteq I_A$, a set of arguments called the input extension, is defined by

$$\mathit{Stb}(\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle, E_{I_A}) = \mathit{stb}(\langle \mathcal{A}_A \cup E_{I_A}, \rightarrow_A \cup (R_{I_A} \cap (E_{I_A} \times \mathcal{A}_A)) \rangle)_{\mathcal{A}_A}$$

Where for a set of extensions S, $S_{\mathcal{A}_A} = \{s \cap \mathcal{A}_A \mid s \in S\}$.

We then give the definition of collective acceptance, which may be seen as the arguments accepted by an external observer.

Definition 7.1.3 (Collective acceptance)**.** The collective stable semantics of multi-agent argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src \rangle$ is the set of extensions $E \subseteq \mathcal{A}$ such that for all agents $A \in Ag$ we have $E_A \in \mathit{Stb}(\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle, E \cap I_A)$. We write $\textbf{\textit{STB}}(\langle \mathcal{A}, \rightarrow, Ag, Src \rangle)$ for the set of all stable extensions of the argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src \rangle$.

Multiagent argumentation is then extended with a social network among the agents, reflecting epistemic trust: an agent trusts another agent if the former accepts the arguments the latter agent accepts. If the social network is reflexive, symmetric and transitive, then network consists of equivalence classes of agents, which may be called coalitions.

Individual and collective acceptance for trust argumentation frameworks is the same as defined before, using trust argumentation frameworks for the individual agents.

Definition 7.1.4 (Trust Argumentation Framework)**.** A Trust argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle$ extends a multiagent argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src \rangle$ with a binary relation $T \subseteq Ag \times Ag$, such that each agent A trusts itself, i.e. $T(A, A)$. We write $T(A)$ for $\{B \mid T(A, B)\}$.

Moreover, the *trust argumentation framework of agent* A is a tuple $\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle$, where $I_A = \{a \in \mathcal{A} \mid Src(a) \in T(A), a \notin \mathcal{A}_A, (a,b) \in \rightarrow, b \in \mathcal{A}_A\}$ is restricted to arguments introduced by agents trusted by agent A.

For individual acceptance we write *StbT*$(\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle, E_{I_A})$ for the set of all stable extensions of the argumentation framework $\langle \mathcal{A}_A, \rightarrow_A, I_A, R_{I_A} \rangle$ defined from the argumentation framework $\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle$ in this way, and each context $E_{I_A} \subseteq I_A$, and for collective acceptance we write *STBT*$(\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle)$ for the set of all stable extensions of the argumentation framework $(\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle)$.

The following definition generalises Dung's stable extensions in terms of sub-frameworks. A stable sub-framework is a sub-framework having exactly one stable extension. A sub-framework semantics called AFRA semantics was introduced by Baroni *et al.* [92] and one called attack semantics by Villata *et al.* [93].

Definition 7.1.5 (Stable sub-frameworks)**.** The framework $\langle \mathcal{A}', \rightarrow' \rangle$ is a *stable sub-framework* of $\langle \mathcal{A}, \rightarrow \rangle$ if and only if $\mathcal{A}' \subseteq \mathcal{A}$, $\rightarrow' \subseteq \rightarrow \cap (\mathcal{A}' \times \mathcal{A}')$, and $\langle \mathcal{A}', \rightarrow' \rangle$ has exactly one stable extension which is also a stable extension of $\langle \mathcal{A}, \rightarrow \rangle$. We write $\textsc{stb}(\langle \mathcal{A}, \rightarrow \rangle)$ for the set of all stable sub-frameworks of the argumentation framework $\langle \mathcal{A}, \rightarrow \rangle$.

Example 7.1.1 below illustrates a multi-agent argumentation with an accused, a witness, a prosecutor and finally a judge who is evaluating collective acceptance. We show how variations in individual conditional acceptance, i.e. variations in what to reveal for the judgement of collective acceptance, can lead to different outcomes, some good and some bad for the accused.

Example 7.1.1. There occurred a murder at Laboratory C, of which Acc is being accused. There are a witness Wit and a prosecutor Prc. Acc has in mind two arguments:

a_1 that he was at Laboratory A on the day of the murder. (This is a fact known to Acc)

a_2 that he is innocent. (This is Acc's claim)

Prc entertains:

a_6 that only Acc could have killed the victim. (This is Prc's claim)

Meanwhile, Wit believes in certain information. He has three arguments:

a_3 Acc stayed at home on the day of the murder, having previously lost his ID card. (Wit originally believes this to be a fact)

a_4 Acc could enter any laboratory provided he is with his ID card. (This is a fact known to Wit)

a_5 Acc could not have been at Laboratory C at the time of the murder. (This is Wit's claim)

The multi-agent argumentation in Figure 7.1 (**A**) represents this example, showing which argument attacks which argument. We denote this multi-agent argumentation by $\langle \mathcal{A}, \mathcal{R}, Ag, Src \rangle$.

In this example, Prc has no reason to drop his/her argument a_6; neither does Acc, seeing no benefit in conceding to a_6, have any reason to drop a_2. Hence, we only consider the contexts: $E_{I_{\text{Acc}}} = E_{I_{\text{Prc}}} = \emptyset$. How Wit responds to the fact known to Acc (a_1), however, can

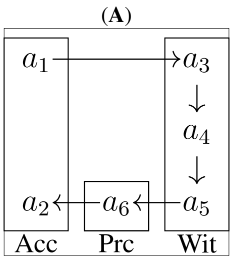


Figure 7.1: Accused (Acc), witness (Wit), and prosecutor (Prc).

prove crucial for Acc to be judged innocent or guilty by the judge who computes collective acceptance.

Case A Suppose $E_{I_{\text{Wit}}} = \emptyset$, which signifies that Wit is either not aware of a_1 or just ignoring it. Then any individual acceptance function of Wit outputs a set of sub-frameworks of $\langle \mathcal{A}, \mathcal{R} \rangle_{\text{Wit}}$ such that they have $\{a_3, a_5\}$ as their only one stable extension. Thus, suppose the acceptance function to output just $\langle \mathcal{A}, \mathcal{R} \rangle_{\text{Wit}}$, as shown in Figure 7.1 (**A**). Then the stable sub-framework of $\langle \mathcal{A}, \mathcal{R}, Ag, Src \rangle$ is $\langle \mathcal{A}, \mathcal{R}, \{a_1, a_4, a_6\} \rangle$. Hence Acc is not judged innocent by the judge.

On the other hand, if the acceptance function of Wit outputs just $\langle \{a_3, a_5\}, \emptyset \rangle$, as shown in Figure 7.1 (**B**). Then the stable sub-framework is $\langle \mathcal{A} \backslash \{a_4\}, \mathcal{R} \backslash (\{a_4\} \times \mathcal{A} \cup \mathcal{A} \times \{a_4\}), \{a_1, a_2, a_5\} \rangle$. Acc is judged innocent by the judge.

Case B Suppose $E_{I_{\text{Wit}}} = \{a_1\}$, which signifies that Wit takes a_1 into account. Then any individual acceptance function of Wit outputs a set of sub-frameworks of $\langle \mathcal{A}, \mathcal{R} \rangle_{\text{Wit}}$ such that they have $\{a_4\}$ as their only one stable extension. Thus, suppose the acceptance function to output just $\langle \{a_4\}, \emptyset \rangle$, as shown in Figure 7.1 (**C**). Then the stable sub-framework is $\langle \mathcal{A} \backslash \{a_3, a_5\}, \mathcal{R} \backslash (\{a_3, a_5\} \times \mathcal{A} \cup \mathcal{A} \times \{a_3, a_5\}), \{a_1, a_4, a_6\} \rangle$. Again, Acc is not judged innocent by the judge.

7.1.2 Dialogue semantics

We apply the same commitment-graph structure to the argumentation dialogue between the agents. The agents first commit to a single extension of their internal framework when multiple ones exist, and then decide how to share it. They may opt to fully share their arguments, exposing counter-arguments that they know about but locally reject, or they may decide to share the arguments they accept without mentioning any of the counter-arguments they are aware of.

In the original framework described in Chapter 3, we provide a framework for the analysis of the commitments made in the process of selecting a single extension from a set of them. The commitments are represented in directed graphs, where the nodes represent commitments made by the agent towards a progressively smaller subset of extensions, until only a single one remains.

When applying this approach to our multi-agent dialogue setting, we will slightly adapt this framework. When it is their turn, the agents either chooses between which arguments to accept, or once that is done, they choose which arguments to share. While they want to share every argument that they accept, this might not be the case for the arguments they reject. For each argument they are aware of but do not accept, the agents have the opportunity to either leave them out, or share them with their peers.

We present a definition of commitments about arguments. We allow the agents to choose for each argument that they reject whether to communicate it or not. We therefore introduce pairs to represent these decisions.

Definition 7.1.6 (Commitments on arguments and coherence)**.** Given an argument a and $c \in \{say, hide\}$, we say that a triple (c, a) is a *commitment on* a. Given a set C of commitments on arguments, we say that C is *coherent* if there is no argument a such that $(say, a) \in C$ and $(hide, a) \in C$.

For simplicity, we write $s(a)$ instead of (say, a) and $h(a)$ instead of $(hide, a)$, and for a set of commitments C, we write C^s for $\{a \mid s(a) \in C\}$ and C^h for $\{a \mid h(a) \in C\}$.

We then define a sub-framework semantics which takes these commitments into account.

Definition 7.1.7 (Committed stable sub-framework semantics)**.** Let $\langle \mathcal{A}, \rightarrow \rangle$ be an argumentation framework and C a coherent set of commitments on arguments in $\mathcal{A}$. We define the *C-committed stable sub-framework semantics* to be:

$$STBC^C(\langle \mathcal{A}, \rightarrow \rangle) = \{\langle \mathcal{A} \setminus C^h, (\rightarrow \cap (\mathcal{A} \setminus C^h)^2) \setminus \mathcal{A} \times \mathcal{E} \rangle \mid \mathcal{E} \in stb(\langle \mathcal{A}, \rightarrow \rangle)\}$$

The C-committed stable semantics first removes all arguments which one has committed to hide. It then looks at the remaining framework, and proceeds to remove the attacks on any arguments from the stable extension. This then forces a single stable extension in the particular framework.

We can now adapt the decision graph structure from Dauphin *et al.* [94] to the triple-A frameworks. So the individual agents still have to commit on which arguments to accept when their internal argumentation allows for multiple extensions, but then they also have to commit on which ones to communicate. Now that we have defined the notion of C-committed stable semantics in order to let an agent choose for each argument he rejects whether to share it or not, we examine the impact these commitments have on the final extensions determined by the overall observer, or in our running examples, the judge.

Definition 7.1.8 (Multi-agent commitment graph)**.** Let $\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle$ be a trust argumentation framework. We say that a labelled directed acyclic graph $(\mathcal{C}, \mathcal{V}, l)$, where $\mathcal{C}$ is a set of sets of commitments about $\mathcal{A}$, $\mathcal{V} \subseteq \mathcal{C} \times \mathcal{C}$ and l is a function assigning labels to both $\mathcal{C}$ and $\mathcal{V}$ as described below, is a *multi-agent commitment graph* for $\langle \mathcal{A}, \rightarrow, Ag, Src, T \rangle$ iff all of the following hold:

1. $\emptyset \in \mathcal{C}$ and every other node can be reached from it via $\mathcal{V}$;

2. the non-leaf[1] nodes are labelled with some $ag \in Ag$;

3. the edges are labelled with decisions of the form $c(a)$, where $a \in \mathcal{A}$ and $c \in \{s, h\}$, and if $l(c_1, c_2) = c(a)$ then $Src(a) = l(c_1)$;

[1] A leaf node is a node with no outgoing edge.

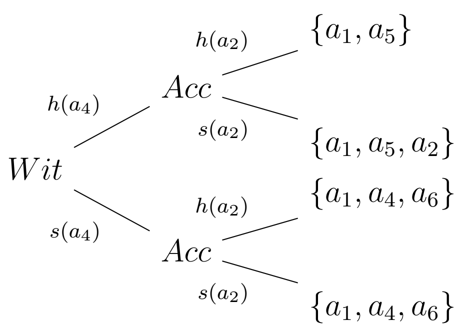


Figure 7.2: Commitment graph for the argumentation framework of Example 7.1.1.

4. for all $e = (c_1, c_2) \in \mathcal{V}$, $c_2 = c_1 \cup l(e)$;
5. for every leaf node c, $l(c) = E$, where E is such that $STB(\langle \mathcal{A}, \rightarrow, Ag, Src \rangle, \{STBC^c\}) = \{(\mathcal{A}', \rightarrow', E)\}$.

The first constraint is that the starting point is with no decision made yet, from which all the considered cases can be reached. The second item, together with item three, force the process to consider only the commitments for a single agent at a time. The third item also represents the fact that a commitment on a certain argument can only be made by the agent who is its source. Item four represents the carrying over of commitments, and the last item labels the nodes with the resulting collective extension.

Example 7.1.2 (Three agents, continued from Example 7.1.1)**.** Consider the scenario described in Example 7.1.1, with the case where Acc trusts Prc and Prc trusts Wit, but Wit does not trust Acc. Wit does not need to condition his local framework, and therefore accepts a_3 and a_5, and rejects a_4. Wit may now decide whether to share a_4 or hide it, in which case he would only communicate the framework $\langle \{a_3, a_5\}, \emptyset \rangle$. Now, Acc considers the possibility that a_6 is acceptable and thus that a_2 is not. Acc can then decide whether to share a_2 or hide, since he considers it might not be acceptable.

7.2 Argument label functions

7.2.1 Introduction

Abstract argumentation frameworks (AFs) [4] are reasoning structures where one aims at extracting sets of jointly acceptable arguments. One of the central methods to do so is the labeling-based approach [13], in which one derives labelings which assign to each argument one of three labels: *in*, *out* or *undec*. The arguments that are labeled *in* represent the arguments that are jointly acceptable, while the arguments that are *out* represent the ones that are defeated by those. The last label, *undec* (*undecided*), represents the cases where one cannot, or decides with proper justification, not to assign either of these two labels. One advantage of the labeling approach is that to verify that an argument is correctly labeled, one only needs to check the labels of its direct ancestors. This allows for a more local evaluation, which is still equivalent to other global approaches such as the extension-based approach.

Many enrichments of abstract argumentation frameworks have been studied, e.g. with bipolar argumentation frameworks which add a second relation of support [48], or with argumentation frameworks with recursive attacks (AFRA) [49] in which attacks may also target other attacks. One methodology for evaluating such enriched frameworks while staying coherent with the basic framework is the flattening approach [56], where the enrichments added to the abstract argumentation frameworks are expressed in terms of extra arguments and attacks, allowing one to evaluate them as abstract argumentation frameworks. An essential concern in the flattening approach is whether the extra arguments and attacks produce the same behavior as the one intended by the enrichment they flatten. This raises a question: Which relations connecting two arguments can be expressed in terms of arguments and attacks alone?

In this chapter we propose to address this research question by studying the representability of label functions, i.e. of functions which map each of the three labels to one of these labels. We prove that in preferred, complete and grounded semantics, eleven label functions can be represented by an AF while sixteen label functions cannot be represented by any AF. We show how this analysis of label functions can be applied to prove an impossibility result: Argumentation frameworks extended with a certain kind of weak attack relation cannot be flattened to the standard Dung argumentation frameworks. Furthermore we also briefly discuss representability of label functions with respect to the stable and semi-stable semantics.

The structure of the section is as follows: in Subsection 7.2.2 we formally define the notion of label function and what it means to represent them as abstract argumentation frameworks. In Subsection 7.2.3 we show which of the twenty-seven label functions are representable and which ones are unrepresentable in the context of the complete, grounded and preferred semantics, and briefly mention the case of the stable semantics. In Subsection 7.2.5 we discuss the implications of these impossibility results for the flattening of a particular relation: a weak attack relation that does not propagate the undecided label. We then discuss related work in Subsection 7.2.6 and future work in Subsection 7.2.7, where we also briefly discuss the case of the semi-stable semantics. We provide a short conclusion in Subsection 7.2.8.

7.2.2 Label Functions

In this section we define the basic notions of a label function, an input-output argumentation framework and the representability of a label function. We write *Labs* for the set of possible labels $\{\mathit{in}, \mathit{out}, \mathit{undec}\}$.

Definition 7.2.1. A *label function* LF is a function from *Labs* to *Labs*.

Definition 7.2.2. Let LF_1 and LF_2 be two label functions. Then $LF_1 \circ LF_2$ denotes the composition of these two label functions that is defined as $LF_1 \circ LF_2(L) = LF_1(LF_2(L))$.

We use the triplet $(LF(\mathit{in}), LF(\mathit{out}), LF(\mathit{undec}))$ to refer to LF in a concise way. For example, the triplet $(\mathit{out}, \mathit{undec}, \mathit{in})$ denotes the label function that maps *in* to *out*, *out* to *undec* and *undec* to *in*.

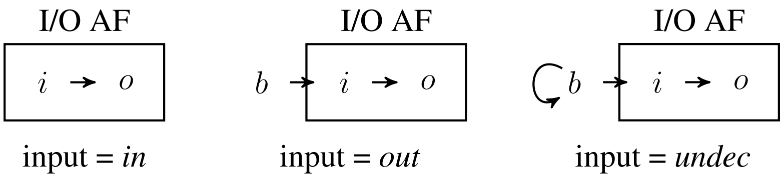


Figure 7.3: The three standard AFs for the I/O AF that cgp-represents the label function $(\textit{out}, \textit{in}, \textit{undec})$.

Definition 7.2.3. An *input-output argumentation framework* (*I/O AF*) is a tuple $(\mathcal{A}, \rightarrow, i, o)$, where $(\mathcal{A}, \rightarrow)$ is an argumentation framework and $i, o \in \mathcal{A}$.

Definition 7.2.4. Given an input-output argumentation framework $G = (\mathcal{A}, \rightarrow, i, o)$, with an argument $b \notin \mathcal{A}$ and a label $L \in \textit{Labs}$, the *standard argumentation framework w.r.t. G and L* – denoted $F_{st}(G, L)$ – is the argumentation framework $(\mathcal{A}', \rightarrow')$, where $\mathcal{A}'$ and $\rightarrow'$ are defined through the following case distinction:

- If $L = \textit{in}$, then $\mathcal{A}' = \mathcal{A}$ and $\rightarrow' = \rightarrow$.
- If $L = \textit{out}$, then $\mathcal{A}' = \mathcal{A} \cup \{b\}$ and $\rightarrow' = \rightarrow \cup \{(b, i)\}$.
- If $L = \textit{undec}$, then $\mathcal{A}' = \mathcal{A} \cup \{b\}$ and $\rightarrow' = \rightarrow \cup \{(b, b), (b, i)\}$.

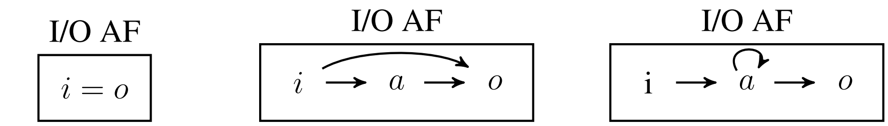
Definition 7.2.5. Let σ be an argumentation semantics. An input-output argumentation framework G *represents* a label function LF w.r.t. σ iff for every $L \in \textit{Labs}$, $\sigma(F_{st}(G, L)) \neq \emptyset$ and for every labeling $\textit{Lab} \in \sigma(F_{st}(G, L))$, $\textit{Lab}(i) = L$ and $\textit{Lab}(o) = LF(L)$.

Definition 7.2.6. Let σ be an argumentation semantics. A label function LF is called σ-*representable* iff there is some input-output argumentation framework G that represents LF w.r.t. σ.

In this work, we shall focus on three of the most well-known semantics, namely complete, grounded and preferred. The principles that these semantics satisfy make them the most appropriate to start with.

Definition 7.2.7. We define cgp to be the set of semantics $\{\textit{complete, grounded, preferred}\}$. If a label function can be σ-represented for every $\sigma \in$ cgp, we say that the function is cgp-representable. Similarly, if a label function cannot be σ-represented for any $\sigma \in$ cgp, we say that the function is cgp-unrepresentable.

Example 7.2.1. Consider the label function $(\textit{out}, \textit{in}, \textit{undec})$ which maps *in* to *out* and vice-versa, leaving *undec* as it is. This function can be cgp-represented as depicted in Fig. 7.3. By having the input directly attack the output, when the input is *in*, it forces the output to be *out*. Conversely, when the input is *out*, there is no attacker of the output left, so it must be *in*. And finally when the input is *undec*, the undecided label propagates to the output.



$(\mathit{in}, \mathit{out}, \mathit{undec})$ $(\mathit{out}, \mathit{out}, \mathit{undec})$ $(\mathit{in}, \mathit{undec}, \mathit{undec})$

Figure 7.4: cgp-representation of three label functions.

Example 7.2.2. Fig. 7.4 depicts three I/O AFs that cgp-represent the label functions $(\mathit{in}, \mathit{out}, \mathit{undec})$, $(\mathit{out}, \mathit{out}, \mathit{undec})$ and $(\mathit{in}, \mathit{undec}, \mathit{undec})$ respectively. Note that the I/O AF that represents the identity function $(\mathit{in}, \mathit{out}, \mathit{undec})$ consists only of a single argument, so that the input argument i and the output argument o are the same argument.

We now define how two input-output argumentation frameworks can be composed into a single one. The intuitive idea is that the output of the first I/O AF is used as input for the second I/O AF.

Definition 7.2.8. Let $G_1 = (\mathcal{A}_1, \rightharpoonup_1, i_1, o_1)$ and $G_2 = (\mathcal{A}_2, \rightharpoonup_2, i_2, o_2)$ be two input-output argumentation frameworks with $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$, and let $c \notin \mathcal{A}_1 \cup \mathcal{A}_2$. Then we define $G_1 \oplus G_2$ to be the input-output argumentation framework $(\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{c\}, \rightharpoonup_1 \cup \rightharpoonup_2 \cup \{(o_1, c)\} \cup \{(c, i_2)\}, i_1, o_2)$.

The following theorem establishes that composed AFs represent composed label functions with respect to the complete, grounded and preferred semantics.

Theorem 7.2.1. *Let LF_1 and LF_2 be representable label functions, and let $G_1 = (\mathcal{A}_1, \rightharpoonup_1, i_1, o_1)$ and $G_2 = (\mathcal{A}_2, \rightharpoonup_2, i_2, o_2)$ be input-output argumentation frameworks that represent LF_1 and LF_2 respectively. Then $G_1 \oplus G_2$ cgp-represents $LF_2 \circ LF_1$.*

Proof. Let $\sigma \in$ cgp. Let $L \in \mathit{Labs}$. Then every σ-labeling of $F_{st}(G_1, L)$ assigns the label $LF_1(L)$ to o_1, and every σ-labeling of $F_{st}(G_2, LF_1(L))$ assigns the label $LF_2 \circ LF_1(L)$ to o_2. We need to show that every σ-labeling of $F_{st}(G_1 \oplus G_2, L)$ assigns the label $LF_2 \circ LF_1(L)$ to o_2. So let Lab be a σ-labeling of $F_{st}(G_1 \oplus G_2, L)$. By the Directionality principle for σ, $\mathit{Lab}|_{F_{st}(G_1, L)} \in \sigma(F_{st}(G_1, L))$, so $\mathit{Lab}(o_1) = LF_1(L)$.

We write $F^* = (\mathcal{A}^*, \rightharpoonup^*)$ for $F_{st}(G_2, LF_1(L))$, and we write $F' = (\mathcal{A}', \rightharpoonup')$ for $F_{st}(G_1 \oplus G_2, L)$. Recall that by Definition 7.2.4, $\mathcal{A}^* = \mathcal{A}_2 \cup \{b\}$ if $LF_1(L) = \mathit{out}$ and $\mathcal{A}^* = \mathcal{A}_2$ otherwise. Let Lab_b be the unique σ-labeling of $F^*|_{\mathcal{A}^* \setminus \mathcal{A}_2}$, i.e. $\mathit{Lab}_b = \{(b, \mathit{in})\}$ if $LF_1(L) = \mathit{out}$, and $\mathit{Lab}_b = \emptyset$ otherwise.

By the SCC-recursiveness principle for σ, there is a base function BF_σ such that for every AF $F = (\mathcal{A}, \rightharpoonup)$, $\sigma(F) = GF(BF_\sigma, F, \mathcal{A})$. In particular, $\sigma(F') = GF(BF_\sigma, F', \mathcal{A}')$, i.e. $\mathit{Lab} \in GF(BF_\sigma, F', \mathcal{A}')$. Clearly $|\mathit{SCCs}(F')| > 1$, so by the SCC-recursiveness principle for σ, for all $S \in \mathit{SCCs}(F')$, we have:

$$\begin{array}{l} \text{There is a } \mathit{Lab}' \in GF(BF_\sigma, F'|_{S \setminus D_{F'}(\mathit{Lab})}, U_{F'}(S, \mathit{Lab})) \\ \text{such that } \mathit{Lab}|_S = \mathit{Lab}' \cup \mathit{Lab}^{\mathit{out}}_{F'|_{S \cap D_{F'}(\mathit{Lab})}}. \end{array} \tag{7.1}$$

When applying this equation in Case 3 below, we will need to make use of the fact that if $S \subseteq \mathcal{A}_2$, then $S \setminus D_{F'}(\mathit{Lab}) = S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$ and $U_{F'}(S, \mathit{Lab}) = U_{F^*}(S, \mathit{Lab}_b \cup$

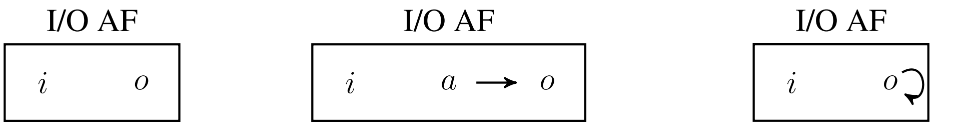
$(\mathit{Lab}|_{\mathcal{A}_2}))$. In the following we show that $U_{F'}(S, \mathit{Lab}) \subseteq U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$. The facts that $U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})) \subseteq U_{F'}(S, \mathit{Lab})$, that $S \setminus D_{F'}(\mathit{Lab}) \subseteq S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$ and that $S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})) \subseteq S \setminus D_{F'}(\mathit{Lab})$ can be established in a similar way.

> Suppose $x \in U_{F'}(S, \mathit{Lab})$, i.e. $x \in S$ and there is no y such that $(y, x) \in \rightarrow'$, $y \not\sim x$ and $\mathit{Lab}(y) \neq \mathit{out}$. Now suppose for a contradiction that $x \notin U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$, i.e. there is a z such that $(z, x) \in \rightarrow^*$, $z \not\sim x$ and $\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})(z) \neq \mathit{out}$. We distinguish two cases:
>
> Case (i): $z \in \mathcal{A}_2$. Choose $y := z$. Note that since $z \not\sim x$, $(z, x) \neq (i, i)$. So $(y, x) = (z, x) \in \rightarrow'$, $y \not\sim x$ and $\mathit{Lab}(y) = \mathit{Lab}(z) = \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})(z) \neq \mathit{out}$. This contradicts the assumption that there is no such y.
>
> Case (ii): $z = b$. In this case $LF_1(L) = \mathit{out}$. Choose $y := c$. Then $(b, x) \in \rightarrow^*$, i.e. $x = i$. Therefore $(y, x) = (c, i) \in \rightarrow'$. Additionally $y \not\sim x$. Furthermore, since $\mathit{Lab}(o_1) = LF_1(L) = \mathit{out}$ and o_1 is the only attacker of c, $\mathit{Lab}(c) = \mathit{in}$, i.e. $\mathit{Lab}(y) \neq \mathit{out}$. This contradicts the assumption that there is no such y.
>
> Thus $U_{F'}(S, \mathit{Lab}) \subseteq U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$, as required.

Recall that every σ-labeling of F^* assigns the label $LF_2 \circ LF_1(L)$ to o_2. We now establish the required result that $\mathit{Lab}(o_2) = LF_2 \circ LF_1(L)$ by showing that $\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}) \in \sigma(F^*)$. By the SCC-recursivity of σ, it is enough to show that $\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}) \in GF(BF_\sigma, F^*, \mathcal{A}^*)$. Clearly $|\mathit{SCCs}(F^*)| > 1$, so by the SCC-recursiveness principle for σ it is enough to show that for all $S \in \mathit{SCCs}(F^*)$, there is a $\mathit{Lab}' \in GF(BF_\sigma, F^*|_{S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))}, U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})))$ such that $(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S = \mathit{Lab}' \cup \mathit{Lab}^{\mathit{out}}_{F^*|_{S \cap D_{F^*}(\mathit{Lab})}}$. So let $S \in \mathit{SCCs}(F^*)$. We distinguish two cases:

Case 1: $S = \{b\}$ and $LF_1(L) = \mathit{out}$. In this case, $(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S$ is $\{(b, \mathit{in})\}$, i.e. the unique σ-labeling of the AF $F^*|_S = (\{b\}, \emptyset)$. By the SCC-recursivity of σ, $GF(BF_\sigma, F^*|_S, S) = \sigma(F^*|_S)$, so $\{(b, \mathit{in})\} \in GF(BF_\sigma, F^*|_S, S)$. Note that S is unattacked, i.e. $F'|_{S \cap D_{F'}(\mathit{Lab})}$ is the empty AF and $\mathit{Lab}^{\mathit{out}}_{F'|_{S \cap D_{F'}(\mathit{Lab})}}$ is the empty labeling. So we can choose $\mathit{Lab}' = (\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S = \{(b, \mathit{in})\}$. Furthermore, $F^*|_{S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))} = F^*|_S$ and $U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})) = S$. Thus $GF(BF_\sigma, F^*|_{S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))}, U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))) = GF(BF_\sigma, F^*|_S, S)$, which contains $\mathit{Lab}' = \{(b, \mathit{in})\}$, as required.

Case 2: $S = \{b\}$ and $LF_1(L) = \mathit{undec}$. In this case, $(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S$ is $\{(b, \mathit{undec})\}$, i.e. the unique σ-labeling of the AF $F^*|_S = (\{b\}, \{b, b)\})$. By the SCC-recursivity of σ, $GF(BF_\sigma, F^*|_S, S) = \sigma(F^*|_S)$, so $\{(b, \mathit{undec})\} \in GF(BF_\sigma, F^*|_S, S)$. Note that S is unattacked, i.e. $F'|_{S \cap D_{F'}(\mathit{Lab})}$ is the empty AF and $\mathit{Lab}^{\mathit{out}}_{F'|_{S \cap D_{F'}(\mathit{Lab})}}$ is the empty labeling. So we can choose $\mathit{Lab}' = (\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S = \{(b, \mathit{undec})\}$. Furthermore, $F^*|_{S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))} = F^*|_S$ and $U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})) = S$. Thus $GF(BF_\sigma, F^*|_{S \setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))}, U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))) = GF(BF_\sigma, F^*|_S, S)$, which contains $\mathit{Lab}' = \{(b, \mathit{undec})\}$, as required.



(*in*,*in*,*in*) (*out*,*out*,*out*) (*undec*,*undec*,*undec*)

Figure 7.5: `cgp`-representation of the three constant label functions.

Case 3: $S \subseteq \mathcal{A}_2$. Then $(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S = \mathit{Lab}|_S$. Furthermore, $S \in \mathit{SCCs}(F')$, so by equation (7.1), there is a $\mathit{Lab}' \in GF(BF_\sigma, F'|_{S\setminus D_{F'}(\mathit{Lab})}, U_{F'}(S, \mathit{Lab}))$ such that $\mathit{Lab}|_S = \mathit{Lab}' \cup \mathit{Lab}^{\mathit{out}}_{F'|_{S\cap D_{F'}(\mathit{Lab})}}$. Thus $(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))|_S = \mathit{Lab}' \cup \mathit{Lab}^{\mathit{out}}_{F'|_{S\cap D_{F'}(\mathit{Lab})}}$, as required. Furthermore, as shown above, $S\setminus D_{F'}(\mathit{Lab}) = S\setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$ and $U_{F'}(S, \mathit{Lab}) = U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))$, so $\mathit{Lab}' \in GF(BF_\sigma, F^*|_{S\setminus D_{F^*}(\mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2}))}, U_{F^*}(S, \mathit{Lab}_b \cup (\mathit{Lab}|_{\mathcal{A}_2})))$, as required. □

The following corollary directly follows from Theorem 7.2.1

Corollary 7.2.2. *If LF_1 and LF_2 are `cgp`-representable, then $LF_1 \circ LF_2$ is `cgp`-representable.*

7.2.3 Representability of Label Functions

In this subsection, we will categorize the twenty seven label functions into eleven functions that are `cgp`-representable and sixteen functions that are not `cgp`-representable.

As we will show below, a label function is `cgp`-representable iff it is either a constant function or maps *undec* to *undec*. This motivates the following definition:

Definition 7.2.9. We define the set `Rep` as the following set of label functions:

$$\texttt{Rep} = \{(\mathit{in}, \mathit{in}, \mathit{in}), (\mathit{out}, \mathit{out}, \mathit{out})\} \cup \{(l, l', \mathit{undec}) \mid l, l' \in \mathit{Labs}\}$$

Theorem 7.2.3. *Every function in `Rep` is `cgp`-representable.*

Proof. We have already given the `cgp`-representations for four of those functions in the Examples 7.2.1 and 7.2.2. We additionally have representations for the three constant functions $(\mathit{in}, \mathit{in}, \mathit{in})$, $(\mathit{out}, \mathit{out}, \mathit{out})$ and $(\mathit{undec}, \mathit{undec}, \mathit{undec})$ in Figure 7.5. The missing four functions can be represented by combinations of these seven as follows:

- $(\mathit{in}, \mathit{in}, \mathit{undec}) = (\mathit{out}, \mathit{in}, \mathit{undec}) \circ (\mathit{out}, \mathit{out}, \mathit{undec})$;
- $(\mathit{undec}, \mathit{in}, \mathit{undec}) = (\mathit{in}, \mathit{undec}, \mathit{undec}) \circ (\mathit{out}, \mathit{in}, \mathit{undec})$;
- $(\mathit{out}, \mathit{undec}, \mathit{undec}) = (\mathit{out}, \mathit{in}, \mathit{undec}) \circ (\mathit{in}, \mathit{undec}, \mathit{undec})$;
- $(\mathit{undec}, \mathit{out}, \mathit{undec}) = (\mathit{out}, \mathit{in}, \mathit{undec}) \circ (\mathit{undec}, \mathit{in}, \mathit{undec})$. □

Aside from the widely used semantics included in the set $\texttt{cgp}$, the stable semantics is another well-known semantics which is also complete-based. Notice however that the stable semantics does not allow for any *undec* arguments, and thus no framework could stable-represent a label function as defined in Def. 7.2.5, since having *undec* as input would automatically mean there is no extension in the corresponding standard AF, so no output could be given. We can however define a similar notion over 2-valued labelings, i.e. restricting the functions to only two possible inputs and outputs: *in* and *out*.

This restriction leaves us with only four different possible label functions, and an interesting small result is that all of these are stable-representable. $(\mathit{out}, \mathit{in})$ is stable-represented by the I/O AF in Figure 7.3 and $(\mathit{in}, \mathit{out})$ by the I/O AF on the left in Figure 7.4. $(\mathit{in}, \mathit{in})$ and $(\mathit{out}, \mathit{out})$ are stable-represented by the I/O AFs in Figure 7.5, respectively on the left and in the middle.

Proposition 7.2.4. *The four 2-valued label functions* $(\mathit{in}, \mathit{out})$, $(\mathit{out}, \mathit{in})$, $(\mathit{in}, \mathit{in})$ *and* $(\mathit{out}, \mathit{out})$ *are all stable-representable.*

7.2.4 Unrepresentable Label Functions

In this subsection we establish that the sixteen labeling functions not included in $\texttt{Rep}$ are actually $\texttt{cgp}$-unrepresentable. We first consider the labeling functions $(\mathit{undec}, \mathit{undec}, \mathit{out})$ and $(\mathit{out}, \mathit{undec}, \mathit{out})$ with respect to the preferred and grounded semantics.

Lemma 7.2.5. *The labeling functions* $(\mathit{undec}, \mathit{undec}, \mathit{out})$ *and* $(\mathit{out}, \mathit{undec}, \mathit{out})$ *are* $\texttt{cgp}$*-unrepresentable.*

Proof. We show this using a proof by contradiction. Assume $G = (\mathcal{A}, \rightarrow, i, o)$ is an input-output argumentation framework that $\texttt{cgp}$-represents either $(\mathit{undec}, \mathit{undec}, \mathit{out})$ or $(\mathit{out}, \mathit{undec}, \mathit{out})$. This means that in every complete, grounded and preferred labeling of $F_{st}(G, \mathit{undec})$, the output argument o is labeled *out*. We first show how to derive a contradiction in the case of preferred. Let *Lab* be a preferred labeling of $F_{st}(G, \mathit{undec})$, and let E be the set of arguments labeled *in* by *Lab*. Since for every admissible set E, there exists a preferred labeling in which the arguments of E are *in* [13], E is admissible w.r.t. $F_{st}(G, \mathit{undec})$. This implies that E is admissible w.r.t. $F_{st}(G, \mathit{out})$:

- Conflict-freeness of E w.r.t. $F_{st}(G, \mathit{out})$ follows from the fact that the only attack in $F_{st}(G, \mathit{out})$ that is not present in $F_{st}(G, \mathit{undec})$ is the attack from the special argument b to the input argument i, but clearly $b \notin E$.

- Self-defence of E w.r.t. $F_{st}(G, \mathit{out})$ follows from the fact that the only attack in $F_{st}(G, \mathit{undec})$ that is not present in $F_{st}(G, \mathit{out})$ is the self-attack on the input argument i, but clearly $i \notin E$.

Now since for every admissible set E, there exists a preferred labeling in which the arguments of E are *in* [13], there exists a preferred labeling Lab' of $F_{st}(G, \mathit{out})$ in which every argument in E is labeled *in*. Since $\mathit{Lab}(o) = \mathit{out}$, some argument c labeled *in* by *Lab* attacks o. But then $c \in E$, so $\mathit{Lab}'(c) = \mathit{in}$, so $\mathit{Lab}'(o) = \mathit{out}$. But this contradicts the assumption that G represents $(\mathit{undec}, \mathit{undec}, \mathit{out})$ or $(\mathit{out}, \mathit{undec}, \mathit{out})$ w.r.t. the preferred semantics, because this would mean that every preferred labeling of $F_{\mathit{st}}(G, \mathit{out})$ labels o as *undec*.

Now we consider the case of the grounded semantics. By a simple transfinite induction one can show that for every ordinal α, $\mathcal{F}^{\alpha}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})(i) = \mathit{undec}$ (since i attacks itself in $F_{\mathit{st}}(G, \mathit{undec})$). Since the grounded labeling is the $\leq$-least fixpoint of $\mathcal{F}_F$ and there exists an ordinal α such that the least fixpoint of $\mathcal{F}_F$ is $\mathcal{F}^{\alpha}_F(\mathit{Lab}_{\mathit{undec}})$, there exists an ordinal α such that the grounded labeling of $F_{\mathit{st}}(G, \mathit{undec})$ is $\mathcal{F}^{\alpha}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$. We now show by transfinite induction that for every ordinal β, $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$:

- $\beta = 0$: Trivial.

- $\beta = 1$: In this case $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ labels as *in* all arguments in $\mathcal{A}$ that are unattacked, and $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$ also labels all these arguments as *in*, and additionally labels the special argument b as *in*. So clearly $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$.

- $\beta = 2$: In this case the *in*-labeled arguments are the same as in the case $\beta = 1$ for both $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ and $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Additionally, every argument attacked by an unattacked argument from $\mathcal{A}$ is labeled *out* by both $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ and $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$, and $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$ additionally labels the special argument i as *out*. So clearly $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$.

- $\beta = \gamma + 1$ for $\gamma \geq 2$: By the inductive hypothesis, we may assume that $\mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. By the definition of $\leq$, every argument labeled *in* by $\mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *in* by $\mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. The fact that every argument in $\mathcal{A}$ has the same attackers in $F_{\mathit{st}}(G, \mathit{out})$ as in $F_{\mathit{st}}(G, \mathit{undec})$ together with the definition of $\mathcal{F}$ imply that every argument in $\mathcal{A}$ that is labeled *out* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *out* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Since $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})(i) = \mathit{undec}$, this implies that every argument that is labeled *out* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *out* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Similarly one can show that every argument that is labeled *in* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *in* by $\mathcal{F}^{\gamma+1}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Thus $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$.

- β is a limit ordinal: Suppose c is an argument such that $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})(c) = \mathit{in}$. This means that there is some $\gamma < \beta$ such that $\mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})(i) = \mathit{in}$. By induction hypothesis, $\mathcal{F}^{\gamma}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}}) = \mathit{in}$, so $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}}) = \mathit{in}$. Thus every argument that is labeled *in* by $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *in* by $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Similarly every argument that is labeled *out* by $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})$ is also labeled *out* by $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$. Thus $\mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\beta}_{F_{\mathit{st}}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$.

Since G represents either $(\mathit{undec}, \mathit{undec}, \mathit{out})$ or $(\mathit{out}, \mathit{undec}, \mathit{out})$ w.r.t. the grounded semantics, the output argument o is labeled *out* by the grounded labeling of $F_{\mathit{st}}(G, \mathit{undec})$.

So $\mathcal{F}^{\alpha}_{F_{st}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}})(o) = \mathit{out}$. But since $\mathcal{F}^{\alpha}_{F_{st}(G,\mathit{undec})}(\mathit{Lab}_{\mathit{undec}}) \leq \mathcal{F}^{\alpha}_{F_{st}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})$, this means that $\mathcal{F}^{\alpha}_{F_{st}(G,\mathit{out})}(\mathit{Lab}_{\mathit{undec}})(o) = \mathit{out}$. So the grounded labeling of $F_{st}(G,\mathit{out})$ labels o as *out*, in contradiction to the assumption that G represents $(\mathit{undec},\mathit{undec},\mathit{out})$ or $(\mathit{out},\mathit{undec},\mathit{out})$ w.r.t. the grounded semantics.

Finally we consider the case of the complete semantics. Every prefered labeling is a complete labeling and every AF has at least one prefered labeling. These two facts together imply that whenever an input-output argumentation framework G represents an labeling function LF w.r.t. the complete semantics, G also represents LF w.r.t. the prefered semantics. So since the labeling functions $(\mathit{undec},\mathit{undec},\mathit{out})$ and $(\mathit{out},\mathit{undec},\mathit{out})$ are not prefered-representable, they are not complete-representable either. □

Now we extend these results to cover all labeling functions not in `Rep`.

Theorem 7.2.6. *The sixteen labeling functions not in* `Rep` *are* `cgp`*-unrepresentable.*

Proof. In Lemma 7.2.5 we have already established that the labeling functions $(\mathit{undec},\mathit{undec},\mathit{out})$ and $(\mathit{out},\mathit{undec},\mathit{out})$ are `cgp`-unrepresentable. For the other fourteen labeling functions not in `Rep` we show this result by showing that if one of them was representable, then one of $(\mathit{undec},\mathit{undec},\mathit{out})$ or $(\mathit{out},\mathit{undec},\mathit{out})$ would be representable too, which would be a contradiction. For this purpose we show in the table below how each of these fourteen labeling functions not in `Rep` or mentioned in Lemma 7.2.5 can be composed with some of the eleven `cgp`-representable labeling functions from `Rep` to define either $(\mathit{undec},\mathit{undec},\mathit{out})$ or $(\mathit{out},\mathit{undec},\mathit{out})$.

The second column of the following table presents for each label function mentioned in the first column a proof in concize notation that shows why the label function in question is `cgp`-unrepresentable. We explain how these proofs in concize notation should be read through the example of the first proof presented in the table: If $(\mathit{in},\mathit{in},\mathit{out})$ were `cgp`-representable, then the fact that $(\mathit{undec},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{undec}) \circ (\mathit{in},\mathit{in},\mathit{out})$ and that $(\mathit{undec},\mathit{out},\mathit{undec})$ is `cgp`-representable would imply that $(\mathit{undec},\mathit{undec},\mathit{out})$ is `cgp`-representable by Corollary 7.2.2, which would contradict Lemma 7.2.5. So we can conclude that $(\mathit{in},\mathit{in},\mathit{out})$ is `cgp`-unrepresentable.

Labeling function	Reason for this labeling function being `cgp`-unrepresentable
—==— $(\mathit{in},\mathit{in},\mathit{out})$	$(\mathit{undec},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{undec}) \circ (\mathit{in},\mathit{in},\mathit{out})$
$(\mathit{in},\mathit{out},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{undec},\mathit{undec}) \circ (\mathit{in},\mathit{out},\mathit{in})$
$(\mathit{in},\mathit{out},\mathit{out})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{undec}) \circ (\mathit{in},\mathit{out},\mathit{out}) \circ (\mathit{out},\mathit{in},\mathit{undec})$
$(\mathit{in},\mathit{undec},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{in},\mathit{undec}) \circ (\mathit{in},\mathit{undec},\mathit{in})$
$(\mathit{in},\mathit{undec},\mathit{out})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{out},\mathit{undec}) \circ (\mathit{in},\mathit{undec},\mathit{out})$
$(\mathit{out},\mathit{in},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{undec},\mathit{undec}) \circ (\mathit{out},\mathit{in},\mathit{in}) \circ (\mathit{out},\mathit{in},\mathit{undec})$
$(\mathit{out},\mathit{in},\mathit{out})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{undec}) \circ (\mathit{out},\mathit{in},\mathit{out})$
$(\mathit{out},\mathit{out},\mathit{in})$	$(\mathit{undec},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{undec},\mathit{undec}) \circ (\mathit{out},\mathit{out},\mathit{in})$
$(\mathit{out},\mathit{undec},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{out},\mathit{undec}) \circ (\mathit{out},\mathit{undec},\mathit{in})$
$(\mathit{undec},\mathit{in},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{in},\mathit{undec}) \circ (\mathit{undec},\mathit{in},\mathit{in}) \circ (\mathit{out},\mathit{in},\mathit{undec})$
$(\mathit{undec},\mathit{in},\mathit{out})$	$(\mathit{undec},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{undec}) \circ (\mathit{undec},\mathit{in},\mathit{out})$
$(\mathit{undec},\mathit{out},\mathit{in})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{out},\mathit{undec}) \circ (\mathit{undec},\mathit{out},\mathit{in}) \circ (\mathit{out},\mathit{in},\mathit{undec})$
$(\mathit{undec},\mathit{out},\mathit{out})$	$(\mathit{out},\mathit{undec},\mathit{out}) = (\mathit{undec},\mathit{out},\mathit{out}) \circ (\mathit{out},\mathit{in},\mathit{undec})$
$(\mathit{undec},\mathit{undec},\mathit{in})$	$(\mathit{undec},\mathit{undec},\mathit{out}) = (\mathit{out},\mathit{in},\mathit{undec}) \circ (\mathit{undec},\mathit{undec},\mathit{in})$

□

7.2.5 Impossibility of Flattening Weak Attacks

Various extensions of argumentation frameworks have been studied in the literature. One fruitful approach to studying such extensions is the flattening methodology, in which extensions of argumentation frameworks are mapped to standard argumentation frameworks through a flattening function that is faithful with respect to the semantics of the extended argumentation frameworks.

In this section we show how the theory of label functions can be used prove impossibility results concerning flattenings of certain extensions of argumentation frameworks.

Multiple authors have considered extending argumentation frameworks with a support relation in addition to an attack relation. Frameworks with both an attack and a support relation are called *bipolar argumentation frameworks* (*BAFs*), and multiple approaches to formalizing their semantics have been studied in the literature, for example *deductive support* [95], *necessary support* [30] and *evidential support* [31]. We briefly sketch the deductive support approach. The intuitive meaning of a deductive support from argument a to argument b is that whenever a is accepted, b must be accepted too. The definition of argumentation semantics for AFs has been adapted to a definition of semantics of BAFs that formalize this intuitive interpretation of deductive support [95]. Later it was shown that the flattening function that replaces every deductive support from a to b by a pair of attacks, namely from b to an auxiliary argument $Z_{(a,b,)}$ and from $Z_{(a,b,)}$ to a, is faithful with respect to these semantics, i.e. that flattening a BAF to an AF and then applying a standard argumentation semantics to the resulting AF gives the same result as directly applying the corresponding deductive support semantics to the BAF [26].

In this section we will study an extension of argumentation frameworks with a weak attack relation. For the formal definition of an extended framework, it is irrelevant whether the second relation that gets added to the standard attack relation is a relation of support or a second attack relation. This motivates the following definitions:

Definition 7.2.10. A *two-relation framework* is a triple $(\mathcal{A}, \rightharpoonup, \mathcal{T})$ such that $\rightharpoonup \subseteq \mathcal{A} \times \mathcal{A}$ and $\mathcal{T} \subseteq \mathcal{A} \times \mathcal{A}$.

Definition 7.2.11. A *two-relation semantics* is a function σ that maps any two-relation framework $B = (\mathcal{A}, \rightharpoonup, \mathcal{T})$ to a set $\sigma(B)$ of labelings of B. The elements of $\sigma(B)$ are called σ-labelings of B.

Definition 7.2.12. Let σ be an argumentation semantics and let σ' be a two-relation semantics. We say that σ' *extends* σ iff for every two-relation framework $B = (\mathcal{A}, \rightharpoonup, \mathcal{T})$ with $\mathcal{T} = \emptyset$, $\sigma'(B) = \sigma((\mathcal{A}, \rightharpoonup))$.

The semantics that have been defined for BAFs with a deductive support relation (and also the semantics for a necessary support relation) extend the corresponding semantics of standard AFs. This fact directly follows from the fact that BAFs can be flattened to AFs in a way that is faithful with respect to these BAF semantics.

An important feature of the flattening of the deductive support relation that we informally sketched above is that every support between two arguments is flattened in an analogous way and the support relation does not have any additional (potentially non-local) effect on the attack relation of the resulting AF. This feature can be formalized as follows:

Definition 7.2.13. Let $B = (\mathcal{A}, \rightarrow, \mathcal{T})$ be a two-relation framework, and let $G = (\mathcal{A}', \rightarrow', i, o)$ be an I/O AF. The *G-flattening of B* is the AF $\mathit{flat}_G(B) = (\mathcal{A}^*, \rightarrow^*)$, where $\mathcal{A}^* := \mathcal{A} \cup \{(a,b,c) \mid (a,b) \in \mathcal{T} \text{ and } c \in \mathcal{A}' \setminus \{i,o\}\}$ and $\rightarrow^* := \rightarrow \cup \{((a,b,c),(a,b,c')) \mid (a,b) \in \mathcal{T}, (c,c') \in \rightarrow' \text{ and } c,c' \notin \{i,o\}\} \cup \{(a,(a,b,c)) \mid (a,b) \in \mathcal{T} \text{ and } (i,c) \in \rightarrow'\} \cup \{((a,b,c),a) \mid (a,b) \in \mathcal{T} \text{ and } (c,i) \in \rightarrow'\} \cup \{(b,(a,b,c)) \mid (a,b) \in \mathcal{T} \text{ and } (o,c) \in \rightarrow'\} \cup \{((a,b,c),b) \mid (a,b) \in \mathcal{T} \text{ and } (c,o) \in \rightarrow'\}$.

Definition 7.2.14. Let σ be an argumentation semantics and let σ' be a two-relation semantics that extends σ. We say that σ' *admits a uniform local flattening w.r.t.* σ iff there exists an I/O AF G such that for every two-relation argumentation framework B, $\sigma'(B) = \sigma(\mathit{flat}_G(B))$.

We now consider a way of interpreting two-relation frameworks in which the second relation is not a support relation, but rather a *weak attack* relation. The intention behind our notion of a weak attack is that when an argument a is weakly attacked by an argument b, one can accept a without being able to defend a against the weak attack from b, but that in all other respects (such as conflict-freeness), weak attacks behave like the standard attacks of abstract argumentation, which we from now on call *strong attacks* to distinguish them clearly from weak attacks. In the labeling-based approach this means that the local effect that weak attacks from arguments with certain labels have on other arguments is generally analogous to the local effect of strong attacks, with the only exception that an argument can be labeled *in* even though it is weakly attacked by an argument labeled *undec*. So we need the following adaptation of Definition 4.2.2 (the abbreviation "s/w" stands for "strong/weak"):

Definition 7.2.15. Let $B = (\mathcal{A}, \rightarrow, \mathcal{T})$ be a two-relation framework, and let *Lab* be a labeling of B.

- An argument $a \in \mathcal{A}$ is called *s/w-legally in w.r.t. Lab* iff every argument that strongly attacks a is labeled *out* by *Lab* and every argument that weakly attacks a is labeled either *out* or *undec*.
- An argument $a \in \mathcal{A}$ is called *s/w-legally out w.r.t. Lab* iff some argument that strongly or weakly attacks a is labeled *in* by *Lab*.
- An argument $a \in \mathcal{A}$ is called *s/w-legally undec w.r.t. Lab* iff no argument that strongly or weakly attacks a is labeled *in* by *Lab* and some argument that strongly attacks a is labeled *undec* by *Lab*.

Now we define the semantics for two-relation frameworks with strong and weak attacks analogously as for standard AFs:

Definition 7.2.16. Let $B = (\mathcal{A}, \rightarrow, \mathcal{T})$ be a two-relation framework, and let *Lab* be a labeling of B.

- *Lab* is an *s/w-complete labeling* of B iff every argument that *Lab* labels *in* is s/w-legally *in* w.r.t. *Lab*, every argument that *Lab* labels *out* is s/w-legally *out* w.r.t. *Lab*, and every argument that *Lab* labels *undec* is s/w-legally *undec* w.r.t. *Lab*.

- *Lab* is an *s/w-grounded labeling* of B iff *Lab* is an s/w-complete labeling of B in which the set of *in*-labeled arguments is minimal w.r.t. set inclusion.

- *Lab* is an *s/w-preferred labeling* of B iff *Lab* is an s/w-complete labeling of B in which the set of *in*-labeled arguments is maximal w.r.t. set inclusion.

One can easily see that these three semantics extend the corresponding semantics of standard AFs.

The following theorem establishes that the weak attack relation cannot be flattened to the strong attack relation in a uniform local way:

Theorem 7.2.7. *Let $\sigma \in cgp$. Then s/w-σ does not admit a uniform local flattening w.r.t. σ.*

Proof. Suppose for a contradiction that s/w-σ does admit a uniform local flattening w.r.t. σ, i.e. there is an I/O AF G such that for every two-relation argumentation framework B, s/w-$\sigma(B) = \sigma(\mathit{flat}_G(B))$.

Consider the following three two-relation frameworks:

$$\begin{aligned} B_{\mathit{in}} &:= (\{i, o\}, \emptyset, \{(i, o)\}) \\ B_{\mathit{out}} &:= (\{i, o, b\}, \{(b, i)\}, \{(i, o)\}) \\ B_{\mathit{undec}} &:= (\{i, o, b\}, \{(b, b), (b, i)\}, \{(i, o)\}) \end{aligned}$$

From Definition 7.2.4, one can easily see that for $L \in \mathit{Labs}$, $F_{st}(G, L) = \mathit{flat}_G(B_L)$ (up to isomorphism; auxiliary arguments may have different names in the two frameworks). Now from Definition 7.2.16, one can easily see that

$$\begin{aligned} \sigma(F_{st}(G, \mathit{in})) &= \text{s/w-}\sigma(B_{\mathit{in}}) = \{\{(i, \mathit{in}), (o, \mathit{out})\}\}, \\ \sigma(F_{st}(G, \mathit{out})) &= \text{s/w-}\sigma(B_{\mathit{out}}) = \{\{(i, \mathit{out}), (o, \mathit{in}), (b, \mathit{in})\}\}, \text{ and} \\ \sigma(F_{st}(G, \mathit{undec})) &= \text{s/w-}\sigma(B_{\mathit{undec}}) = \{\{(i, \mathit{undec}), (o, \mathit{in}), (b, \mathit{undec})\}\}. \end{aligned}$$

So G represents $(\mathit{out}, \mathit{in}, \mathit{in})$ w.r.t. σ, contradicting Theorem 7.2.6. □

7.2.6 Related Work

In the work of Baroni et al. [53], a similar methodology is introduced, where argumentation frameworks are partitioned, allowing for partitions to be evaluated locally. This local evaluation function needs to condition on the potential statuses of attackers from outside the partition, but does not need to consider the whole rest of the framework. From their results on decomposability of semantics, one could derive a result similar to our Theorem

7.2.1 but restricted to finite argumentation frameworks. We however chose to consider infinite argumentation frameworks as well in our work, as it grants more weight to the unrepresentability result derived in Section 7.2.3.

The work of Rienstra et al. [14] considers the partitioning of argumentation frameworks such that different semantics are applied to different partitions. In these cases, when evaluating the acceptance status of arguments within a partition, only the outside arguments which are the source of an attack targeting an argument inside that partition need to be considered, using a similar input/output methodology.

Enrichments of argumentation frameworks, such as the AFRA [49] and the BAF [48] have been interpreted in some cases using a flattening approach [26, 56] which expresses higher-level relations in terms of auxiliary arguments and attacks, which can replace the original relation in a local fashion. Our results would prove useful when devising flattenings for existing or future enrichments, or showing no such flattening is possible.

7.2.7 Future Work

In future work, one could generalize the concept of a label function by dropping the requirement that the output argument always has the same label; these generalized label functions would therefore have a set of possible labels as their output value. Additionally one could drop the distinction between input argument and output argument, thus allowing an external effect on both arguments and looking at the set of label pairs that these two arguments may take over the different extensions. This would yield to a generalized theory of binary relations between arguments that have a local effect expressible in the 3-label approach. While there are only 27 label functions, the number of such different relations between arguments is 2^{36}, so the classification according to their representability is likely to be much more complex. Such a classification would allow one to extend the impossibility result from Section 7.2.3 to other enrichments of abstract argumentation frameworks, or provide insights on how to flatten new enrichments.

Another line of future work would be to investigate the representability with respect to other semantics such as semi-stable [96], stage [17], stage2 [97], CF2 [18], and the more recent SCF2 [22] and weakly complete [98]. We briefly briefly present some preliminary findings for representability with respect to the semi-stable semantics.

The semi-stable semantics [96] has been often criticized for not satisfying a number of standard principles such as directionality [10]. There is however the interesting fact that some functions which are $\mathtt{cgp}$-unrepresentable turn out to be semi-stable-representable.

Example 7.2.3. Consider the I/O AF depicted in Figure 7.6. When the input argument i_1 is labeled *in*, the output argument o_1 is forced to be *out*, since it is directly attacked by i_1. In the case of *undec* input, o_1 cannot be *in*, and by the nature of the semi-stable semantics which minimizes the *undec* labeling, the only option is to have a being *in* and thus o_1 is again *out*. Lastly we consider the case that the input is *out*: While in this case the preferred semantics would give us two labelings with either o_1 or a being *in*, the semi-stable semantics will produce a single labeling with the o_1 labeled *in*, because this way the label *undec* can be avoided completely, even for the self-attacking argument d. So this I/O AF represents the $(\mathit{out}, \mathit{in}, \mathit{out})$ function, which is $\mathtt{cgp}$-unrepresentable.

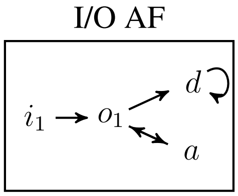
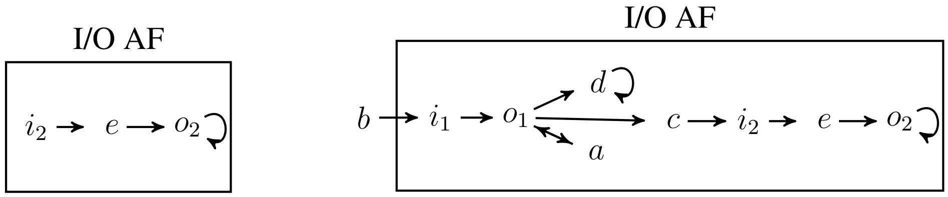


Figure 7.6: A semi-stable representation of the cgp-unrepresentable function (*out*, *in*, *out*).



(*undec*,*out*,*undec*)

Composed I/O AF with *out* as input.

Figure 7.7: Failure of composition with semi-stable semantics. When the I/O AF from Figure 7.6 is composed with the I/O AF on the left, we obtained the I/O AF depicted on the right, which however does not produce a consistent label for o_2 when the input is *out*.

While the semi-stable semantics might be able to represent more functions, the fact that the semi-stable semantics does not satisfy the directionality principle (see [10]) brings about other issues, notably a lack of compositionality, as illustrated by a counter-example in Figure 7.7. On the left, we have an I/O AF which semi-stable-represents the function (*undec*, *out*, *undec*). When composed with the I/O AF from Figure 7.6, we obtain the framework on the right. In that composed framework, when the input is *out*, we now obtain two extensions, namely one where o_1 is *in*, and one where a is *in*. This second extension is now possible, because when o_1 is *in*, we have o_2 labeled *undec*, while when a is *in*, we have d labeled *undec*, so both of these options minimize the set of *undec*-labeled arguments. Since these two labelings have different labels for o_2, this composed I/O AF does not represent any labeling function.

Since our methodology of characterising labeling functions via composition would not work in the case of semi-stable, we leave such an analysis for future work.

7.2.8 Conclusion

In this section, we formally introduce argumentation label functions, and address the question of which functions are representable with an argumentation framework, focusing on the complete, grounded and preferred semantics, for which the labeling approach has been widely studied. We provide a proof that two representations of label functions can be composed to yield the composed label function, and use this finding to categorize the twenty seven label functions into eleven label functions that are representable and sixteen that are unrepresentable with respect to these three semantics. We also briefly investigate the case of the stable semantics, which is quite straightforward since it only allows for two different labels. We then discuss how the label function approach can be used to prove an impossibility result about the flattening approach for enrichments of abstract argumentation frameworks. We briefly investigate the case of the semi-stable semantics, as it allows for the representation of some functions which are not representable with respect to the other semantics. However due to the non-directional nature of the semantics, the composability result does not hold, hindering generalizations as done for the other semantics.

Chapter 8

Conclusion

This thesis addresses the following research questions:

- What can be gained by refining the process of selecting an extension from a set of extensions?
- How can this refined structure allow us to combine different semantics?
- How can we evaluate acceptability in a framework combining many enrichments?
- How can arguments be constructed for an enriched framework with more than just an attack relation?

With respect to the first research question, we have proposed a methodologically novel approach to choosing extensions of argumentation frameworks by studying abstract and concrete commitment graphs that correspond to step-wise commitments about the choice of extension. Inspired by the principle-based approach to abstract argumentation, we have studied two principles that mappings from AFs to commitment graphs should satisfy. We have presented preliminary results in combining this approach with multi-agent argumentation and dialogue semantics.

For the second research question, we introduce a dynamic approach to combine two argumentation semantics to yield a third one, based on the step-wise construction introduced for the first research question. In particular, we provide a formal environment for the analysis of step-wise relations between labeled framework with an increase in the label precision, whose reachable fixpoints correspond to some standard direct semantics. We define and discuss two approaches to combining two given update relations to yield a third update relation, an approach based on algorithmically motivated update relations and an approach based on *merging* maximally fine-grained update relations. For both approaches, we examine how to obtain update relations for the complete labeling by combining update relations for the preferred and grounded labelings. Furthermore, we have defined novel semantics using the merge approach, including a semantics that meaningfully combines features of naive-based and complete-based semantics.

Regarding the third research question, we have examined several extensions of abstract argumentation frameworks that add higher-order explanatory features, recursive attacks, necessary and deductive support, and an incompatibility relation, all allowed to originate and target sets of arbitrary elements. In the cases of higher-order attacks and support of both kinds, we have presented a flattening function, which allows us to instantiate these extended framework as standard AFs. We have shown that in the case of AFRAs, the complete semantics defined in terms of the flattening is equivalent to the complete semantics which has been defined directly on AFRAs. We have then aggregated these extensions into one framework, EEAFs, defined a labeling semantics for it, and then alternatively defined the semantics in terms of its flattening to EAFs, showing the two are equivalent. Finally, we have explored an application of EEAFs to argumentation from a research-level philosophy book.

Finally, for the fourth research question, we have presented ASPIC-END, a structured argumentation framework which allows for the construction of explananda and explanations between arguments. Additionally, the framework allows for the construction of arguments using hypothetical reasoning, where one may introduce assumptions which are not necessarily present in the knowledge base, and upon fulfilling proper conditions can results to conclusions which may have consequences outside the scope of these assumptions.

This thesis explores different aspects of the meaning of acceptability in argumentation, on the one hand by refining the process of selecting a single extension from a set of them via a step-wise commitment process, later using this construction to combine different semantics. On the other hand this thesis examines how enrichments of abstract argumentation frameworks, introduced to provide more expressivity and new criteria for the acceptability of arguments, can often still be expressed in terms of arguments and attacks alone with the introduction of auxiliary elements, even when multiple such enrichments are combined in a single framework.

References

[1] D. T. McRuer, D. Graham, and I. Ashkenas, *Aircraft dynamics and automatic control*. Princeton University Press, 2014.

[2] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, “Robotic surgery: A current perspective,” *Annals of surgery*, vol. 239, no. 1, p. 14, 2004.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artificial intelligence*, vol. 77, no. 2, pp. 321–357, 1995.

[5] V. Lifschitz, *Answer set programming*. Springer, 2019.

[6] S. Modgil and H. Prakken, “The ASPIC+ framework for structured argumentation: a tutorial,” *Argument & Computation*, vol. 5, no. 1, pp. 31–62, 2014.

[7] P. M. Dung, R. A. Kowalski, and F. Toni, “Assumption-based argumentation,” in *Argumentation in artificial intelligence*, Springer, 2009, pp. 199–218.

[8] A. J. García and G. R. Simari, “Defeasible logic programming: An argumentative approach,” *Theory and practice of logic programming*, vol. 4, no. 1+ 2, pp. 95–138, 2004.

[9] P. Baroni, M. Caminada, and M. Giacomin, “Abstract argumentation frameworks and their semantics,” in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds., College Publications, 2018.

[10] L. van der Torre and S. Vesic, “The principle-based approach to abstract argumentation semantics,” in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds., College Publications, 2018.

[11] P. Baroni and M. Giacomin, “On principle-based evaluation of extension-based argumentation semantics,” *Artificial Intelligence*, vol. 171, no. 10, pp. 675–700, 2007, Argumentation in Artificial Intelligence.

[12] S. Doutre and J.-G. Mailly, “Quantifying the Difference between Argumentation Semantic,” in *6th International Conference on Computational models of argument (COMMA 2016)*, vol. 287, Potsdam, Germany, Sep. 2016, pp. 255–262.

[13] P. Baroni, M. Caminada, and M. Giacomin, “An introduction to argumentation semantics,” *The Knowledge Engineering Review*, vol. 26, no. 4, pp. 365–410, 2011.

[14] T. Rienstra, A. Perotti, S. Villata, D. M. Gabbay, and L. van der Torre, “Multi-sorted argumentation,” in *International Workshop on Theorie and Applications of Formal Argumentation*, Springer, 2011, pp. 215–231.

[15] R. Arisaka, K. Satoh, and L. van der Torre, “Anything you say may be used against you in a court of law,” in *Artificial Intelligence and the Complexity of Legal Systems (AICOL)*, Springer, 2018.

[16] M. Giacomin, “Handling Heterogeneous Disagreements Through Abstract Argumentation,” in *International Conference on Principles and Practice of Multi-Agent Systems*, Springer, 2017, pp. 3–11.

[17] B. Verheij, “Two approaches to dialectical argumentation: Admissible sets and argumentation stages,” in *Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop*, Universiteit, 1996, pp. 357–368.

[18] P. Baroni, M. Giacomin, and G. Guida, “SCC-recursiveness: a general schema for argumentation semantics,” *Artificial Intelligence*, vol. 168, no. 1, pp. 162–210, 2005.

[19] S. A. Gaggl and W. Dvořák, “Stage semantics and the SCC-recursive schema for argumentation semantics,” *Journal of Logic and Computation*, vol. 26, no. 4, pp. 1149–1202, 2016.

[20] M. Cramer and M. Guillaume, “Empirical cognitive study on abstract argumentation semantics,” *Frontiers in Artificial Intelligence and Applications*, 2018.

[21] ——, “Empirical study on human evaluation of complex argumentation frameworks,” in *European Conference on Logics in Artificial Intelligence*, Springer, 2019, pp. 102–115.

[22] M. Cramer and L. van der Torre, “SCF2 – an argumentation semantics for rational human judgments on argument acceptability,” *Proceedings of the 8th Workshop*

on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI and Kognition (KIK-2019), 2019.

[23] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, "Encompassing attacks to attacks in abstract argumentation frameworks," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer, 2009, pp. 83–94.

[24] D. M. Gabbay, "Fibring argumentation frames," *Studia Logica*, vol. 93, no. 2-3, pp. 231–295, 2009.

[25] C. Cayrol and M. Lagasquie-Schiex, "Bipolarity in argumentation graphs: Towards a better understanding," *IJAR*, vol. 54, no. 7, pp. 876–899, 2013.

[26] G. Boella, D. M. Gabbay, L. W. van der Torre, and S. Villata, "Support in Abstract Argumentation.," *COMMA*, vol. 216, pp. 111–122, 2010.

[27] D. Šešelja and C. Straßer, "Abstract argumentation and explanation applied to scientific debates," *Synthese*, vol. 190, no. 12, pp. 2195–2217, 2013.

[28] P. Thagard, "Coherence, truth, and the development of scientific knowledge," *Philosophy of science*, vol. 74, no. 1, pp. 28–47, 2007.

[29] J. Dauphin and M. Cramer, "ASPIC-END: Structured Argumentation with Explanations and Natural Deduction," in *Proceedings of the 2017 International Workshop on Theory and Applications of Formal Argument*, 2017.

[30] F. Nouioua and V. Risch, "Argumentation frameworks with necessities," in *International Conference on Scalable Uncertainty Management*, Springer, 2011, pp. 163–176.

[31] N. Oren and T. J. Norman, "Semantics for evidence-based argumentation," *Computational Models of Argument*, 2008.

[32] Y. Wu and M. Caminada, "A Labelling-Based Justification Status of Arguments," *Studies in Logic*, vol. 3, no. 4, pp. 12–29, 2010.

[33] G. Brewka and S. Woltran, "Abstract dialectical frameworks," in *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, 2010.

[34] D. E. Knuth, *The Art of Computer Programming: Volume 1: Fundamental Algorithms*. Addison-Wesley Professional, 1997.

[35] J. L. Gross, J. Yellen, and P. Zhang, *Handbook of graph theory*. Chapman and Hall/CRC, 2013.

[36] Y. Xu and C. Cayrol, “Initial sets in abstract argumentation frameworks,” *Journal of Applied Non-Classical Logics*, pp. 1–20, 2018.

[37] B. Liao, L. Jin, and R. C. Koons, “Dynamics of argumentation systems: A division-based method,” *Artificial Intelligence*, vol. 175, no. 11, pp. 1790–1814, 2011.

[38] B. Liao, *Efficient Computation of Argumentation Semantics*. Academic Press, 2013.

[39] P. Baroni, M. Giacomin, and B. Liao, “Locality and modularity in abstract argumentation,” in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds., College Publications, 2018.

[40] P. Baroni and M. Giacomin, “Solving semantic problems with odd-length cycles in argumentation,” in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer, 2003, pp. 440–451.

[41] W. Dvořák and S. A. Gaggl, “Stage semantics and the SCC-recursive schema for argumentation semantics,” *Journal of Logic and Computation*, vol. 26, no. 4, pp. 1149–1202, 2016.

[42] M. E. Johnson, L. M. Moore, and D. Ylvisaker, “Minimax and maximin distance designs,” *Journal of statistical planning and inference*, vol. 26, no. 2, pp. 131–148, 1990.

[43] S. Modgil and H. Prakken, “Abstract rule-based argumentation,” in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds., College Publications, 2018.

[44] A. C. Kakas and P. Moraitis, “Argumentation based decision making for autonomous agents,” in *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*, ACM, 2003, pp. 883–890, ISBN: 1-58113-683-8.

[45] Y. Dimopoulos, P. Moraitis, and L. Amgoud, “Extending Argumentation to Make Good Decisions,” in *Algorithmic Decision Theory, First International Conference, ADT 2009, Venice, Italy, October 20-23, 2009. Proceedings*, F. Rossi and A. Tsoukiàs, Eds., ser. Lecture Notes in Computer Science, vol. 5783, Springer, 2009, pp. 225–236, ISBN: 978-3-642-04427-4.

[46] L. Amgoud and H. Prade, “Using arguments for making and explaining decisions,” *Artif. Intell.*, vol. 173, no. 3-4, pp. 413–436, 2009.

[47] W. Dvorák, M. Järvisalo, J. P. Wallner, and S. Woltran, “Complexity-Sensitive Decision Procedures for Abstract Argumentation (Extended Abstract),” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*

2015, Buenos Aires, Argentina, July 25-31, 2015, Q. Yang and M. Wooldridge, Eds., AAAI Press, 2015, pp. 4173–4177, ISBN: 978-1-57735-738-4.

[48] C. Cayrol and M.-C. Lagasquie-Schiex, "On the acceptability of arguments in bipolar argumentation frameworks," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer, 2005, pp. 378–389.

[49] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, "AFRA: Argumentation framework with recursive attacks," *International Journal of Approximate Reasoning*, vol. 52, no. 1, pp. 19–37, 2011.

[50] P. Baroni and M. Giacomin, "On principle-based evaluation of extension-based argumentation semantics," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 675–700, 2007.

[51] L. Amgoud, J. Ben-Naim, D. Doder, and S. Vesic, "Acceptability semantics for weighted argumentation frameworks," 2017.

[52] J. Dauphin and C. Schulz, "Arg Teach – A Learning Tool for Argumentation Theory," in *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, IEEE, 2014, pp. 776–783.

[53] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. van der Torre, and S. Villata, "On the Input/Output behavior of argumentation frameworks," *Artificial Intelligence*, vol. 217, pp. 144–197, 2014.

[54] R. Baumann and G. Brewka, "Expanding argumentation frameworks: Enforcing and monotonicity results.," *COMMA*, vol. 10, pp. 75–86, 2010.

[55] S. Coste-Marquis, S. Konieczny, J.-G. Mailly, and P. Marquis, "On the revision of argumentation systems: Minimal change of arguments statuses.," *KR*, vol. 14, pp. 52–61, 2014.

[56] G. Boella, D. M. Gabbay, L. van der Torre, and S. Villata, "Meta-argumentation modelling I: Methodology and techniques," *Studia Logica*, vol. 93, no. 2-3, pp. 297–355, 2009.

[57] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. van der Torre, and S. Villata, "On the Input/Output behavior of argumentation frameworks," *Artif. Intell.*, vol. 217, pp. 144–197, 2014.

[58] H. Field, *Saving Truth from Paradox*. Oxford University Press, 2008.

[59] S. H. Nielsen and S. Parsons, "A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments," in *International Workshop on Argumentation in Multi-Agent Systems*, Springer, 2006, pp. 54–73.

[60] S. Gottifredi, A. Cohen, A. J. García, and G. R. Simari, "Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations," *Artificial Intelligence*, vol. 262, pp. 336–368, 2018.

[61] A. Cohen, S. Gottifredi, A. J. García, and G. R. Simari, "On the Acceptability Semantics of Argumentation Frameworks with Recursive Attack and Support," in *Computational Models of Argument - Proceedings of COMMA 2016*, 2016, pp. 231–242.

[62] G. Flouris and A. Bikakis, "A comprehensive study of argumentation frameworks with sets of attacking arguments," *International Journal of Approximate Reasoning*, vol. 109, pp. 55–86, 2019.

[63] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni, "Introduction to structured argumentation," *Argument & Computation*, vol. 5, no. 1, pp. 1–4, 2014.

[64] P. Baroni, M. Caminada, and M. Giacomin, "An introduction to argumentation semantics," *The Knowledge Engineering Review*, vol. 26, no. 4, pp. 365–410, 2011.

[65] G. Moore, *Zermelo's axiom of choice: its origins, development, and influence*, ser. Studies in the history of mathematics and physical sciences. Springer-Verlag, 1982, ISBN: 9780387906706.

[66] S. Modgil and H. Prakken, "A general account of argumentation with preferences," *Artificial Intelligence*, vol. 195, pp. 361–397, 2013.

[67] B. Fitelson and E. N. Zalta, "Steps toward a computational metaphysics," *Journal of Philosophical Logic*, vol. 36, no. 2, pp. 227–247, 2007.

[68] E. Zalta, *Abstract objects: An introduction to axiomatic metaphysics*. Springer Science & Business Media, 2012, vol. 160.

[69] C. Benzmüller and B. Woltzenlogel Paleo, "The Inconsistency in Gödel's Ontological Argument: A Success Story for AI in Metaphysics," in *IJCAI 2016*, S. Kambhampati, Ed., vol. 1-3, AAAI Press, 2016, pp. 936–942, ISBN: 978-1-57735-770-4.

[70] C. Benzmüller, L. Weber, and B. Woltzenlogel Paleo, "Computer-Assisted Analysis of the Anderson-Hájek Controversy," *Logica Universalis*, vol. 11, no. 1, pp. 139–151, 2017.

[71] J. L. Pollock, "Defeasible reasoning," *Cognitive science*, vol. 11, no. 4, pp. 481–518, 1987.

[72] ——, *Cognitive carpentry: A blueprint for how to build a person*. Mit Press, 1995.

[73] H. Prakken, "An abstract framework for argumentation with structured arguments," *Argument & Computation*, vol. 1, no. 2, pp. 93–124, 2010.

[74] M. Caminada, S. Modgil, and N. Oren, "Preferences and Unrestricted Rebut," in *Computational Models of Argument - Proceedings of COMMA 2014*, 2014, pp. 209–220.

[75] M. Caminada and L. Amgoud, "On the evaluation of argumentation formalisms," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 286–310, 2007.

[76] M. Beirlaen, J. Heyninck, and C. Straßer, "Reasoning by Cases in Structured Argumentation," in *Proceedings of SAC/KRR 2017*, 2017, pp. 989–994.

[77] ——, "A critical assessment of Pollock's work on logic-based argumentation with suppositions," in *Proceedings of the 17th International Workshop on Non-Monotonic Reasoning*, Forthcoming., 2018.

[78] W. N. Reinhardt, "Some remarks on extending and interpreting theories with a partial predicate for truth," *Journal of Philosophical Logic*, vol. 15, no. 2, pp. 219–251, 1986.

[79] S. Feferman, "Reflecting on incompleteness," *The Journal of Symbolic Logic*, vol. 56, no. 01, pp. 1–49, 1991.

[80] J. Beall, M. Glanzberg, and D. Ripley, "Liar Paradox," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Winter 2016, Metaphysics Research Lab, Stanford University, 2016.

[81] M. Cramer and J. Dauphin, *Technical online appendix to "A Structured Argumentation Framework for Modeling Debates in the Formal Sciences"*, 2018.

[82] E. Zermelo, "Beweis, daß jede Menge wohlgeordnet werden kann," *Mathematische Annalen*, vol. 59, no. 4, pp. 514–516, 1904.

[83] G. Peano, "Additione," *Revista de mathematica*, vol. 8, pp. 143–157, 1906.

[84] J. Hadamard, R Baire, H Lebesgue, and E Borel, "Cinq lettres sur la théorie des ensembles," *Bulletin de la Société mathématique de France*, vol. 33, pp. 261–273, 1905.

[85] E. Zermelo, "Neuer Beweis für die Möglichkeit einer Wohlordnung," *Mathematische Annalen*, vol. 65, no. 1, pp. 107–128, 1907.

[86] G. Priest, *In Contradiction: A Study of the Transconsistent*. Oxford University Press, 2006.

[87] M. W. A. Caminada, W. A. Carnielli, and P. E. Dunne, "Semi-stable semantics," *Journal of Logic and Computation*, vol. 22, no. 5, pp. 1207–1254, 2012. eprint: `/oup/backfile/content\_public/journal/logcom/22/5/10.1093/logcom/exr033/2/exr033.pdf`.

[88] Y. Wu, "Between argument and conclusion-argument-based approaches to discussion, inference and uncertainty," Ph.D. dissertation, University of Luxembourg, Luxembourg, Luxembourg, 2012.

[89] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: a proof assistant for higher-order logic*. Springer Science & Business Media, 2002, vol. 2283.

[90] J. Harrison, "HOL Light: An Overview.," in *TPHOLs*, Springer, vol. 5674, 2009, pp. 60–66.

[91] B. Liao, "Toward incremental computation of argumentation semantics: A decomposition-based approach," *Annals of Mathematics and Artificial Intelligence*, vol. 67, no. 3-4, pp. 319–358, 2013.

[92] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, "AFRA: argumentation framework with recursive attacks," *International Journal of Approximate Reasoning*, vol. 52, no. 1, pp. 19–37, 2011.

[93] S. Villata, G. Boella, and L. W. N. van der Torre, "Attack semantics for abstract argumentation," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, T. Walsh, Ed., IJCAI/AAAI, 2011, pp. 406–413, ISBN: 978-1-57735-516-8.

[94] J. Dauphin, M. Cramer, and L. van der Torre, "Abstract and concrete decision graphs for choosing extensions of argumentation frameworks," *Computational Models of Argument*, 2018.

[95] C. Cayrol and M.-C. Lagasquie-Schiex, "Coalitions of arguments: A tool for handling bipolar argumentation frameworks," *International Journal of Intelligent Systems*, vol. 25, no. 1, pp. 83–109, 2010.

[96] M. Caminada, "Semi-stable semantics," in *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, IOS Press, 2006, pp. 121–130.

[97] W. Dvořák and S. A. Gaggl, “Stage semantics and the SCC-recursive schema for argumentation semantics,” *Journal of Logic and Computation*, vol. 26, no. 4, pp. 1149–1202, 2014.

[98] R. Baumann, G. Brewka, and M. Ulbricht, “Revisiting the foundations of abstract argumentation–semantics based on weak admissibility and weak defense,” in *AAAI Conference on Artificial Intelligence*, 2020.