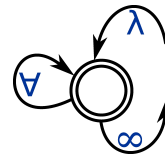




FAKULTÄT  
FÜR INFORMATIK  
Faculty of Informatics



# Argumentation Theory

## Nonmonotonic Reasoning

### BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

### Bachelor of Science

in

Software- and Information Engineering

by

**Christoph Groß, 1025119**

to the Faculty of Informatics  
at the Vienna University of Technology

Advisor: Ao.Prof.Dipl.-Ing.Dr.techn. Christian Georg Fermüller

# Contents

<b>Statutory Declaration</b>	<b>3</b>
<b>Acknowledgements</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>1 Introduction</b>	<b>6</b>
1.1 Nonmonotonic Reasoning . . . . .	6
1.2 Argumentation Theory . . . . .	6
1.2.1 Argumentation Frameworks . . . . .	7
1.2.2 Properties of (Sets of) Arguments . . . . .	7
Admissible Extensions . . . . .	7
Preferred Extensions . . . . .	8
Stable Extensions . . . . .	9
Complete Extensions . . . . .	9
Grounded Extensions . . . . .	9
Semi-stable Extension . . . . .	9
Ideal Extension . . . . .	9
1.2.3 Examples of different Extensions . . . . .	10
<b>2 Conclusions and Implications</b>	<b>14</b>
2.1 Relations between different Extensions . . . . .	14
2.2 Properties of Argumentation Frameworks . . . . .	15
2.2.1 Well-Founded Argumentation Frameworks . . . . .	15
2.2.2 Coherent Argumentation Frameworks . . . . .	16
<b>3 Argumentation Theory and Nonmonotonic Reasoning</b>	<b>17</b>
3.1 Default Logic . . . . .	17
3.2 Default Logic & Argumentation Frameworks . . . . .	18
<b>4 Further Topics</b>	<b>20</b>
4.1 Related Subjects . . . . .	20
4.2 Real-life Applications . . . . .	20
<b>5 Documentation - KAFFEE</b>	<b>21</b>

# List of Figures

1.1	graph of AF (example 1.2)	10
1.2	graph of AF (example 1.3)	10
1.3	graph of AF (example 1.4)	11
1.4	semi-stable but not stable (example 1.5)	11
1.5	ideal extension (example 1.6)	12
1.6	relations of extensions (example 1.7)	13
2.1	hierarchy of extension types	14
5.1	Workflow in KAFFEE	22
5.2	Tab "Argumentation Framework"	22
5.3	Tab "Extensions"	23
5.4	Tab "Step By Step"	23

# Statutory Declaration

Hereby I assure that this thesis is a result of my personal work and that no other than the indicated aids have been used for its completion.

Furthermore I assure that all the quotations and statements, that have been inferred literally or in a general manner from published or unpublished writings, are marked as such.

Beyond this I assure that the work has not been used, neither completely nor in parts, to pass any previous examination.

Vienna, May 16, 2015

---

Christoph Groß

# Acknowledgements

$D := \{Friends, Family, University, christoph\}$

$Friends = \{Max, Pia, Lea, Tobi, Johannes\},$

$Family = \{Sabine, Gerald, Angelika, Johann, Johanna, Patrick\},$

$University = \{Prof. Fermüller\}$

$$\forall x \in D : (proofread(x) \vee listenToWhining(x) \vee help(x)) \implies thanks(christoph, x)$$

# Abstract

Throughout this thesis, I am going to explore the applications of argumentation theory in general and particularly in the context of nonmonotonic reasoning.

First off, I will introduce the necessary formal definitions, give a few easy examples as clarification and draw the theoretical conclusions.

Then I try to show the link between argumentation frameworks and default theory as an example of how argumentation theory can be used for dealing with nonmonotonic reasoning.

In the next step, the reader receives information about things that have not been dealt with in this thesis, some examples of real-life applications and also pointers to the needed sources to get a deeper understanding of the subject.

Finally I will append the documentation of an application specifically written during the course of this thesis, which purpose is to graphically help understanding the subject.

This thesis, including all the cited sources and the application (source code as well as an executable jar file) can be found on [github](#). Should any questions or ideas for improvement arise, do not hesitate to write to [e1025119@student.tuwien.ac.at](mailto:e1025119@student.tuwien.ac.at)

# Chapter 1

## Introduction

### 1.1 Nonmonotonic Reasoning

When talking about nonmonotonic reasoning, one is referring to the process of deducing knowledge within a formal logic, whose consequence relation does not fulfill the requirements of monotonicity.

**Elaboration 1.1** (Monotonicity)

*If for a given set of formulas  $\psi$  and a formula  $A$  the logical consequence  $\psi \vdash A$  holds, then  $\psi, B \vdash A$  still holds.*

As one can see, nonmonotonic sets of formulas might change their deducable knowledge as new facts are added.

### 1.2 Argumentation Theory

In a human context, we (hopefully) argue to reach a sound conclusion. We do that by claiming statements based on premisses, and usually, due to different perception and other factors, these statements tend to vary in their degree of believability.

Argumentation Theory, from a logical point of view, tries to formalize these strategies and introduces rules to that purpose. Therefore arguments are modeled as formulas which might or might not attack each other. For instance, the most believable argument out of two, so to say the "winner", would be the one, that attacks the other argument, and at the same time, does not get attacked by said argument.

In the following chapter, I will provide all the necessary definitions to achieve the above mentioned aim.

### 1.2.1 Argumentation Frameworks

**Definition 1.1** (Argumentation Framework)

$AF := \langle AR, att \rangle$  ■

An argumentation framework AF is defined as a tuple, consisting of a set of arguments and a set of attack relations. The following example should help illustrating this construct a little better.

**Example 1.1**

$s = \text{"Superman is the best superhero!"}$ ,  $b = \text{"Batman is way cooler.."}.$

$AR = \{s, b\}$ ,  $att = \{(b, s)\}$  ■

Since the only tuple in  $att$  is the one that represents  $b$  attacking  $s$ , the (frankly obvious) outcome is that  $b$  is the winning argument.

### 1.2.2 Properties of (Sets of) Arguments

**Definition 1.2** (Conflict-freeness)

A set  $S$  of arguments is said to be conflict-free, if for any argument  $a \in S$ , there is no other argument  $b \in S$  so that  $(a, b) \in att$ . ■

**Definition 1.3** (Acceptability)

An argument  $a \in AR$  is acceptable with respect to a set  $S$ , iff  $S$  attacks any argument  $b \in AR$  that attacks  $a$ . ■

**Remark**

If for said argument  $a$  is not being attacked by any  $b \in AR$ , then it is acceptable with even the empty set. ■

**Definition 1.4** (Admissibility)

A conflict-free Set  $S$  is said to be admissible, iff each argument  $a \in S$  is acceptable with respect to  $S$ . Often,  $S$  is also called an admissible extension. ■

These are the basic properties that can be fulfilled by arguments or sets of arguments. Based on them, we will now introduce a few more interesting definitions. For instance, argumentation theory knows a number of different extensions, which can be used to model the degree of "credulity" of an agent.

Before we discuss those extensions and their implications on argumentation theory, I would like to make a detour and discuss the conventional meaning of extensions in predicate logic.



**Elaboration 1.2** (Extensions)

For any  $n$ -ary predicate  $\Phi$ , its extension can be written as  $\{(x_1, \dots, x_n) | \Phi(x_1, \dots, x_n)\}$ .  
 So, the extension of a predicate is nothing other than the maximum set of tuples fulfilling said predicate. ■

The aforementioned extensions, adapted to fit the needs of argumentation theory, use this idea in combination with certain selective criteria to model the desired behaviour. The first extension takes a rather gullible approach to things; the so called *preferred extension* is defined as follows:

**Remark**

From here on, we will use  $E_i$  to denote an arbitrary but fixed extension of type  $i$ , while  $E_I$  denotes the set of all extensions of type  $i$ .

Also, if we want to talk about the set of all extensions of one specific type, we will use  $i(AF)$ . ■

**Definition 1.5** (Preferred Extensions)

$E_p$  is a preferred extension of  $AF$ , iff  $E_p$  is a maximum admissible set of  $AF$ . ■

As you can see, the preferred extension is made up of the biggest, conflict-free set of arguments, that can be defended against all other arguments not in the set.

The following lemma will be useful later on.

**Lemma 1.1**

For every  $AF$ , there exists at least one preferred extension. ■

**Proof**

The proof will be depending on two different properties of admissible sets:

1. The empty set is always admissible.
2. Given an admissible Set  $S$  and an argument  $a$ , which is acceptable with respect to  $S$ , the unification  $S \cup a$  is also admissible.

(1) directly follows from the definition 1.4

(2) We just have to show, that  $S \cup a$  is conflict-free. To achieve that, we assume the contrary:

There exists an argument  $b \in S$  so that either  $(a, b)$  or  $(b, a)$  holds. There has to be an argument  $c \in S$ , so that either  $(c, a)$  or  $(c, b)$ . We know, that  $S$  is conflict-free, so  $(c, b) \implies \text{false}$

On the other hand,  $c$  and  $a$  are both acceptable with respect to  $S$ ;  $(c, a) \implies \text{false}$ . ■

Another approach would be the *stable extension*, which is defined as follows:

**Definition 1.6** (Stable Extensions)

The conflict-free set  $E_s$  is a stable extension of  $AF$ , iff  $E_s$  attacks each argument  $a \notin E_s$ ; which is equivalent to  $\{a \mid a \text{ is not attacked by } E_s\}$ . ■

Apparently, the stable extension contains all arguments that it does not attack itself. Therefore it has to be a conflict-free, admissible set.

We will explore the differences between those extensions in a later chapter.

For now, there are two more types of extensions, which have to be introduced.

**Definition 1.7** (Complete Extensions)

The admissible set  $E_c$  is a complete extension of  $AF$ , iff every argument which is acceptable with respect to  $E_c$ , is included in  $E_c$ . ■

It is easy to see that many different complete extensions can exist, depending on  $E_c$ . That leads directly to the *grounded extension*.

**Definition 1.8** (Grounded Extensions)

The intersection of all complete extensions is called the grounded extension.

$E_g := \bigcap \text{comp}(AF)$  ■

All those types of extensions can be originally found in [Dung], the more intuitive version of the grounded extension was found in [EgGaWo]. Next, we will take a look at two more extensions, one originally introduced in [CaCaDu], the other found in [DuMaTo].

**Definition 1.9** (Semi-stable Extensions)

A set  $E_{ss}$  is a semi-stable extensions, if it is a complete extension and the set  $E_{ss} \cup \{a \in AR \mid E_{ss} \text{ attacks } a\}$  is maximal (with respect to set inclusion). ■

**Definition 1.10** (Ideal Extensions)

A set  $E_I$  is an ideal extension, if it is the maximal admissible set, that is contained in every preferred extension. ■

Since all the necessary, basic definitions of argumentation frameworks have been laid out, I will go on and provide the promised examples.

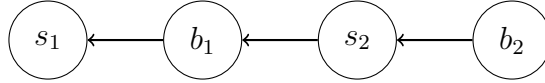
### 1.2.3 Examples of different Extensions

By expanding example 1.1 with a few more arguments, we now have to consider the following example.

#### Example 1.2

$s_1 = \text{"Superman is the best superhero!"}$ ,  $b_1 = \text{"Batman is way cooler.."}$ ,  
 $s_2 = \text{"But Superman is stronger."}$ ,  $b_2 = \text{"Batman possesses genius-level intellect."}$   
 $AR = \{s_1, s_2, b_1, b_2\}$ ,  $att = \{(b_1, s_1), (s_2, b_1), (b_2, s_2)\}$  ■

This argumentation framework could also be depicted by following graph.



**Figure 1.1:** graph of AF (example 1.2)

The extensions of this example would be

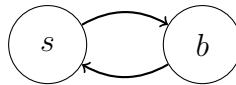
- $E_P = E_S = E_C = E_G = \{\{b_1, b_2\}\}$

The next example is going to be a little more interesting in its outcome.

#### Example 1.3

$s = \text{"Superman was born a hero."}$ ,  $b = \text{"Batman trained hard to become a hero."}$   
 $AR = \{s, b\}$ ,  $att = \{(s, b), (b, s)\}$  ■

This examples graph looks like this.



**Figure 1.2:** graph of AF (example 1.3)

Following extensions can be calculated

- $E_P = E_S = \{\{s\}, \{b\}\}$
- $E_C = \{\{\emptyset\}, \{s\}, \{b\}\}$
- $E_G = \{\{\emptyset\}\}$

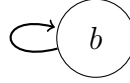
The following example (on the contrary to the above) has different preferred and stable extensions.

**Example 1.4**

$b = \text{"Batman is Clark Kent."}$

$AR = \{b\}$ ,  $att = \{(b, b)\}$  ■

The corresponding graph looks like follows.



**Figure 1.3:** graph of AF (example 1.4)

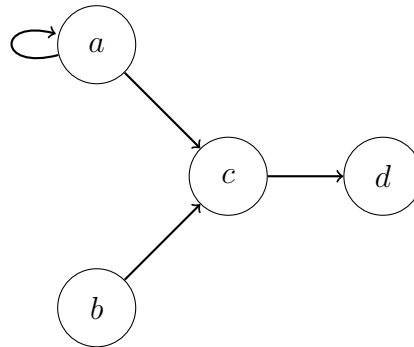
For this example, the extensions would look as follows.

- $E_P = E_C = E_G = \{\{\emptyset\}\}$
- $E_S = \{\emptyset\}$

**Example 1.5**

Another good example, specifically showing that not every semi-stable extension is also stable, was provided in [CaCaDu].

In the following graph, the set  $E_{SS} = \{b, d\}$  is semi-stable but not stable. ■

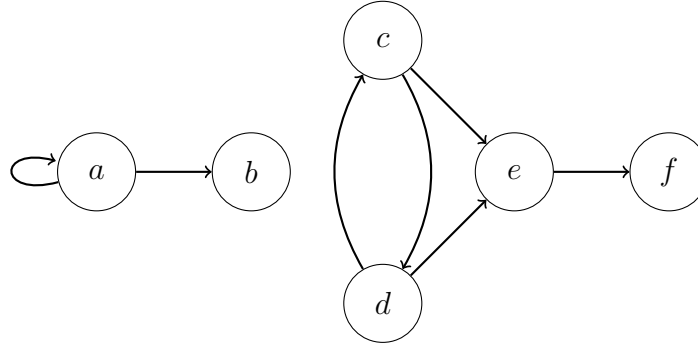


**Figure 1.4:** semi-stable but not stable (example 1.5)

**Example 1.6**

The following example illustrates ideal extensions (and how they differ from e.g. grounded extensions).

In the following graph, the set  $E_I = \{b\}$  is an ideal extension. This example can also be found in [DuMaTo]. ■



**Figure 1.5:** ideal extension (example 1.6)

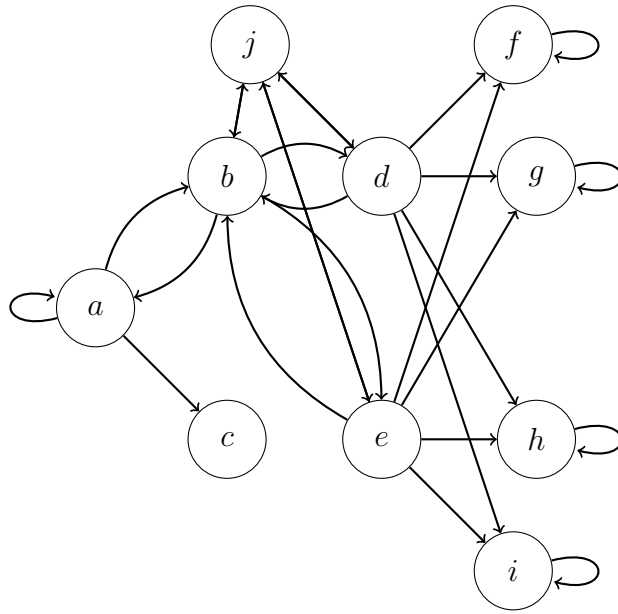
As we can see, the ideal extension  $E_I = \{b\}$  is a subset of every preferred extension  $(\{b, cf\}, \{b, d, f\})$ .

### Example 1.7

Last but not least, we are going to take a look on a slightly more complex example, that gives a deeper insight in the relations between the extensions. It should be considered along with **Figure 2.1**. In the graph below, following different extensions can be observed:

- semi-stable extension  $E_{ss} = \{d, e\}$
- preferred extension  $E_p = \{b, c\}$
- complete extension  $E_c = \{j\}$
- grounded extension  $E_g = \emptyset$

■



**Figure 1.6:** relations of extensions (example 1.7)

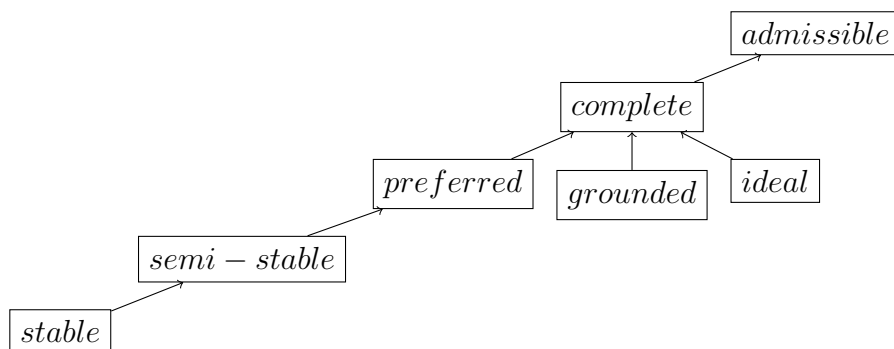
# Chapter 2

## Conclusions and Implications

Now that we have established the basic definitions, it is time to travel further down the rabbit hole and explore the meaning of what we just defined.

### 2.1 Relations between different Extensions

The following, simple tree graph (borrowed from [HuGo]) shows the hierarchy of extension types better than words ever could describe it.



**Figure 2.1:** hierarchy of extension types - An arrow from X to Y denotes that an X-extension is also a Y-extension.

Now I want to show a few specific relations between those aforementioned extensions, since they will be reused in later proofs.

### Lemma 2.1

1. Every preferred extension is a complete extension, but not vice versa.
2. The ground extension is the least (with respect to set inclusion) complete extension

■

#### Proof (1)

It is obvious that every preferred extension (maximum admissible set) is a complete extension. So we only need to show that the other way round is not true.

Assumption: Every complete extension is also a preferred extension.

Let  $AR = \{a, b\}$  and  $att = \{(a, b), (b, a)\}$ .

Then it is easy to see, that  $E_P = \{\{a\}, \{b\}\}$  but  $E_C = \{\emptyset, \{a\}, \{b\}\}$

$\implies \exists E_c : E_c \in E_C \wedge E_c \notin E_P \implies \text{!} \blacksquare$

#### Proof (2)

Proof for this lemma follows directly from definition 1.8. ■

## 2.2 Properties of Argumentation Frameworks

Up to now, we have only talked about sets of arguments and their properties. Now we will focus on the whole argumentation framework and make a connection between its properties and extensions.

### 2.2.1 Well-Founded Argumentation Frameworks

The first new property establishes a coherence between preferred, grounded and stable extensions.

#### Definition 2.1 (Well-Founded Argumentation Frameworks)

An argumentation framework is considered well-founded, iff there exists no infinit sequence of arguments  $A_0, A_1, \dots, A_n, \dots$  where each  $A_{i+1}$  attacks  $A_i$ . ■

#### Remark

This infinite sequence of arguments can be formed by a finite set of arguments (for instance  $AR = \{a, b\}$ ,  $att = (a, b), (b, a)$ ). ■

Although it might not be the most intuitive observation, following lemma can be extracted from this definition.



**Lemma 2.2**

Every well-founded AF has exactly one complete extension, which is also a preferred, grounded and stable extension. ■

**Proof**

Assumption:  $\exists \mathbf{AF} : \mathbf{AF}$  is well-founded and  $\mathbf{E}_p = \mathbf{E}_c = \mathbf{E}_g \notin \mathbf{E}_s$ . We further assume, that  $\exists \mathbf{a} \in \mathbf{AR} \setminus \mathbf{E}_g$  and  $\nexists \mathbf{b} \in \mathbf{E}_g : (\mathbf{b}, \mathbf{a}) \in \mathbf{att}$ .

If  $\mathbf{a}$  is not acceptable with respect to  $\mathbf{E}_g$ , this means that there has to be another argument  $\mathbf{c} \in \mathbf{AR} \setminus \mathbf{E}_g$ , which is also not attacked by  $\mathbf{E}_g$ .

Therefore an infinite sequence of arguments like  $\mathbf{a}$  and  $\mathbf{c}$  has to exist  $\implies \bot$ . ■

**2.2.2 Coherent Argumentation Frameworks**

If an AF has a preferred extension, which is not a stable one too, that indicates some sort of "anomaly" in the framework (see example 1.4).

So what we want to do next, is to find out what criteria has to be met so that preferred and stable extension coincide.

**Definition 2.2** (Coherent Argumentation Frameworks)

An AF is called coherent, if  $\mathbf{E}_p = \mathbf{E}_s$ . ■

**Remark**

It follows directly from the definition, that there always exists at least one stable extension in a coherent AF. ■

# Chapter 3

## Argumentation Theory and Nonmonotonic Reasoning

One reason why argumentation frameworks are important and useful is the fact that they can easily be used to model other logical constructs, often making working with those less complex. For instance, the nonmonotonic "Default Logic", which is briefly described in the following section.

### 3.1 Default Logic

In classical logic, it is not possible to conclude new knowledge from uncertain/unknown information. One scenario for instance, would be the modeling of a default property, which could be assigned to any object as long as we do not have any contradicting facts. This is where default logic comes in handy.

**Definition 3.1** (Default Logic)

*A default theory is a pair  $\langle W, D \rangle$ .  $W$  is the knowledge base, consisting of logical formulas,  $D$  is a set of default rules. Each default rule has following form:*

$$\frac{\textit{Prerequisite} : \textit{Justification}_1, \dots, \textit{Justification}_n}{\textit{Conclusion}}$$

■

A simple (and maybe overused) example is the following.

**Example 3.1**

$$D = \frac{\textit{Bird}(\textit{tweety}) : \textit{Flies}(\textit{Tweety})}{\textit{Flies}(\textit{Tweety})}$$

$$W_1 = \{\textit{Bird}(\textit{tweety})\}$$

$$W_2 = \{\textit{Bird}(\textit{tweety}), \textit{Penguin}(\textit{tweety}), \textit{Penguin}(X) \implies \neg \textit{Flies}(X)\} \blacksquare$$

As we can see, as long as no facts, that negate one of the justifications, can be deduced from our knowledge base, we can infer the conclusion that Tweety can fly (which would be the case with  $\mathbf{W}_1$ ).

On the other hand (with  $\mathbf{W}_2$ ), we could deduce  $\neg \mathbf{Flies}(\mathbf{Tweety})$ , therefore the Justifications would not be met and the default would be overruled by the more important fact.

Furthermore we have to define the so-called "Reiter's extension" (or R-extension).

### Definition 3.2

A R-extension of a default theory  $\langle \mathbf{W}, \mathbf{D} \rangle$  is a set of formulas  $\mathbf{E}_R$ , that satisfies following conditions:

- $\mathbf{E}_R = \bigcup \{ \mathbf{W}_i \mid i \text{ is a natural number} \}$
- $\mathbf{W}_0 = \mathbf{W}$
- $\mathbf{W}_{i+1} = \mathbf{Th}(\mathbf{W}_i) \cup \{ w \mid \exists (p : j_1, \dots, j_k / w) \text{ in } \mathbf{D} \text{ so that } \{ j_n \} \cup \mathbf{E} \text{ consistent for } k \geq n \geq 1, \text{ and } p \in \mathbf{W}_i \}$

$\mathbf{Th}(T)$  denotes the first order closure of a theory  $T$ . ■

The interested reader may want to check [Reiter], where these ideas were originally introduced.

## 3.2 Default Logic & Argumentation Frameworks

As [Dung] points out, a default theory  $T = \langle \mathbf{W}, \mathbf{D} \rangle$  can be interpreted as an argumentation framework  $\mathbf{AF}_d = \langle \mathbf{AR}, \mathbf{Att} \rangle$ .

From here on out, we let  $\mathbf{just}(\mathbf{D})$  denote the set of justifications of a default theory.

The following two criteria are necessary to build the interpretation:

- $\mathbf{AF}_d = \{ (K, k) \mid K \subseteq \mathbf{just}(\mathbf{D}) \}$ , so that  $K$  is a support of the well-formed, closed formula  $k$  with respect to  $T$
- $(K, k)$  attacks  $(K', k')$  iff  $\neg k \in K'$

The following lemma shows that this interpretation is not just a coincidence.

### Lemma 3.1

Let  $\mathbf{E}_a$  be an admissible subset of  $\mathbf{AF}_d$  and  $\mathbf{H} = \bigcup \{ K \mid (K, k) \in \mathbf{E}_a \}$ . Then  $T, \mathbf{H}$  holds, iff  $T$  is consistent. ■

### Proof

From left to right, this should be obvious, so we are only going to take a look at the second part of the proof. To achieve this, we assume the contrary:

There exists a subset  $K$  of  $\mathbf{H}$ , so that  $T, K \vdash \perp$ . Thus, for every  $k$ ,  $(K, k) \in \mathbf{AR}$ . Also, let  $(K', k') \in \mathbf{E}_a$  so that  $K'$  is not empty. Now  $(K, k)$  represents an attack on  $(K', k')$ .

Given the admissibility of  $\mathbf{E}_a$ , there has to be some argument  $a = (H', h') \in \mathbf{E}_a$ , so that  $\neg h' \in K$ ; in other words, since  $(K, k), (K', k') \in \mathbf{E}_a$  and the admissibility of  $\mathbf{E}_a$ , the set can't be conflict-free  $\implies \text{!}$  ■

Now that we have established the possibility to interpret default theories as argumentation frameworks, we want to examine this relationship more closely. There is a correspondence between the R-extensions of a default theory and the stable extensions of an argumentation framework. To show this, we need some more definitions:

### Definition 3.3

Let  $\mathbf{S}$  be a first order theory and  $\mathbf{S}'$  be a set of arguments of  $\mathbf{AF}_d$ . Then, we define:

- $arg(\mathbf{S}) = \{(K, k) \in AR \mid \forall j \in K : \{j\} \cup \mathbf{S} \text{ is consistent}\}$
- $flat(\mathbf{S}') = \{k \mid \exists (K, k) \in \mathbf{S}'\}$

■

Together with the definition of the R-extension, the following lemmata become obvious:

### Lemma 3.2

If  $\mathbf{T}$  is a default theory and  $\mathbf{E}$  is a first order theory. Then,  $\mathbf{E}$  is a R-extension of  $\mathbf{T}$  iff  $\mathbf{E} = flat(arg(\mathbf{E}))$ . ■

### Lemma 3.3

$\mathbf{T} = \langle \mathbf{W}, \mathbf{D} \rangle$  be a default theory,  $\mathbf{E}$  a R-extension of  $\mathbf{T}$  and  $\mathbf{E}'$  a stable extension of  $\mathbf{AF}_d$ . Then

1.  $arg(\mathbf{E})$  is a stable extension of  $\mathbf{AF}_d$
2.  $flat(\mathbf{E}')$  is a R-extension of  $\mathbf{T}$

■

Apparently, the preferred extension semantics of  $\mathbf{AF}_d$  generalizes the R-extension semantics of  $\mathbf{T}$ . Also it captures the intuitive idea of default logic more naturally and thus makes it a bit easier to model and understand. This was just one of the many applications of argumentation frameworks, the interested reader could either get more examples in [Dung] or head on to the "Further Topics" chapter for more information.

# Chapter 4

## Further Topics

To anyone who made it this far into the paper, congratulations.

At this point, some interested reader might have noticed that we have been dealing with the abstract concept of argumentation frameworks almost exclusively. Since this is a bachelor's thesis, I had to focus on one single aspect in order to not get lost in the process of writing it.

Therefore, the idea of this fourth chapter is providing you with pointers to related subjects, sources of deeper understanding and real-life applications to tamper with.

### 4.1 Related Subjects

A very interesting thing is the structure of arguments and how they are extracted. We only used arguments as abstract objects with no underlying structure. If we were trying to solve real problems by implementing them as an argumentation framework, this step would be a necessary one though.

To gain more insight on the structure of arguments, one could read [BeHu] or [Prakken].

If the reader is more interested in how arguments are extracted from knowledge, a process which ideally follows certain rules, then [Pollock] could be a good source.

### 4.2 Real-life Applications

A question that I myself have heard several times from parents, friends and anyone else kind enough to listen to my rambling about argumentation frameworks, is

*"What's the point of all this? How can you use that in real life?"*.

I usually just quoted a professor from my university to get them off my back (*"As a university that is oriented towards fundamental research, that question should not even come up."*), but since this thesis is written to benefit software engineers too, this is a valid question and deserves some clarification.

# Chapter 5

## Documentation - KAFFEE

As mentioned in the abstract, a software tool was written to accompany the reader through this paper and provide the necessary environment to playfully learn more about the several extensions and their relationships.

The **KAFFEE** Argumentation Framework For Engineers and Enthusiasts offers following functions:

- Defining new argumentation frameworks
- Import prepared examples
- Select different types of extensions
- Visualize whole AFs as well as subsets
- Step-by-Step function that explains how an admissible set is created

There are a few built-in shortcuts that make the usage of **KAFFEE** a little easier:

**ALT+h** ..... Help menu

- **ALT+g** ... link to the github project containing all the sources, code etc
- **ALT+c** ... feel free to email me

**ALT+e** ..... Examples menu

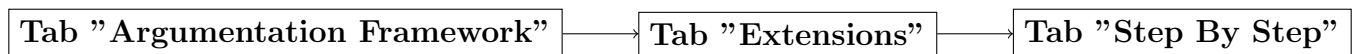
- **ALT+2** ... import example 1.2
- **ALT+3** ... import example 1.3
- **ALT+4** ... import example 1.4
- **ALT+s** ... import example "Step By Step"

The general idea of how to use **KAFFEE** is simple. The tool is made up of three tabs, which are sequentially dependant on the one before. That requires the user to work in tab #1 first, the "Argumentation Framework" tab.

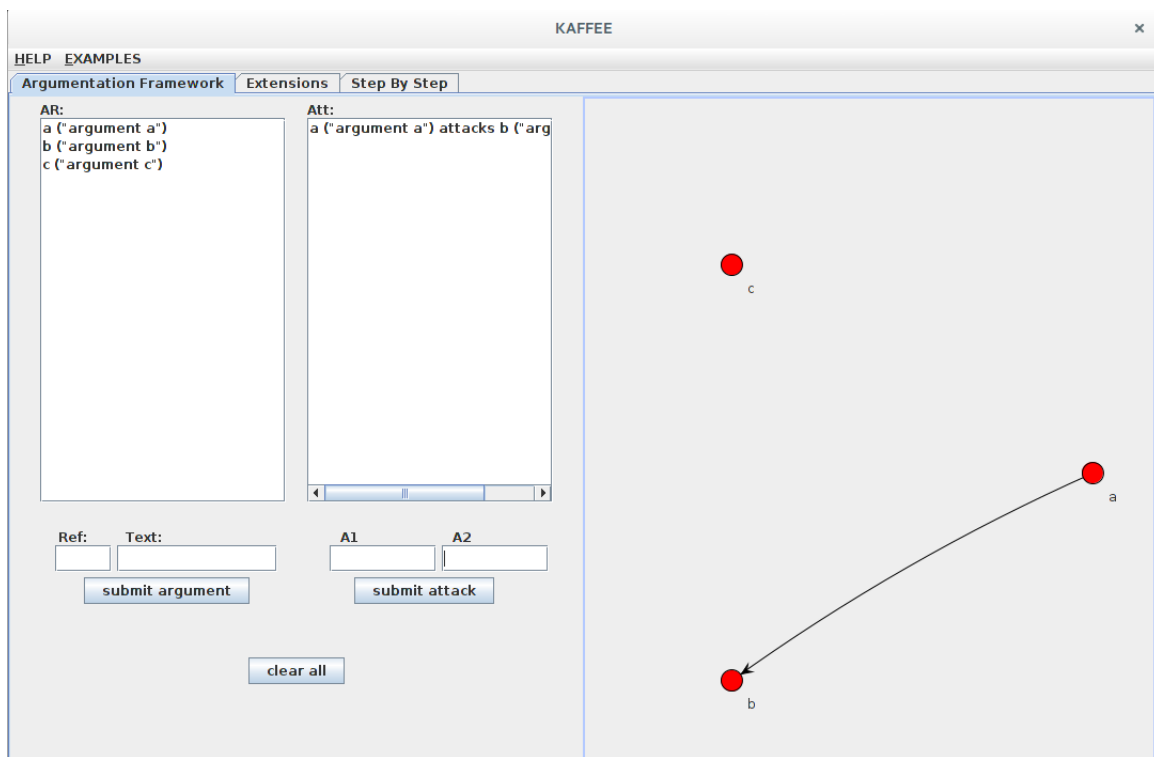
Here, the user has to create/import an argumentation framework. Once that is done, the user can either have a look at the visual representation of his/her framework, or continue and switch to the "Extensions" tab. In this tab, one can choose from a list of possible extensions to be calculated for the given AF. This time, the visual representation on the right will provide coloring of the selected subset, depending on the chosen extension.

The last (and optional) tab is the "Step By Step" tab. Here, one can get a better explanation, in a walkthrough kind of style, of how admissible sets are created.

One iteration while working with **KAFFEE** could be depicted by following graph.



**Figure 5.1:** Workflow in KAFFEE



**Figure 5.2:** Tab "Argumentation Framework"

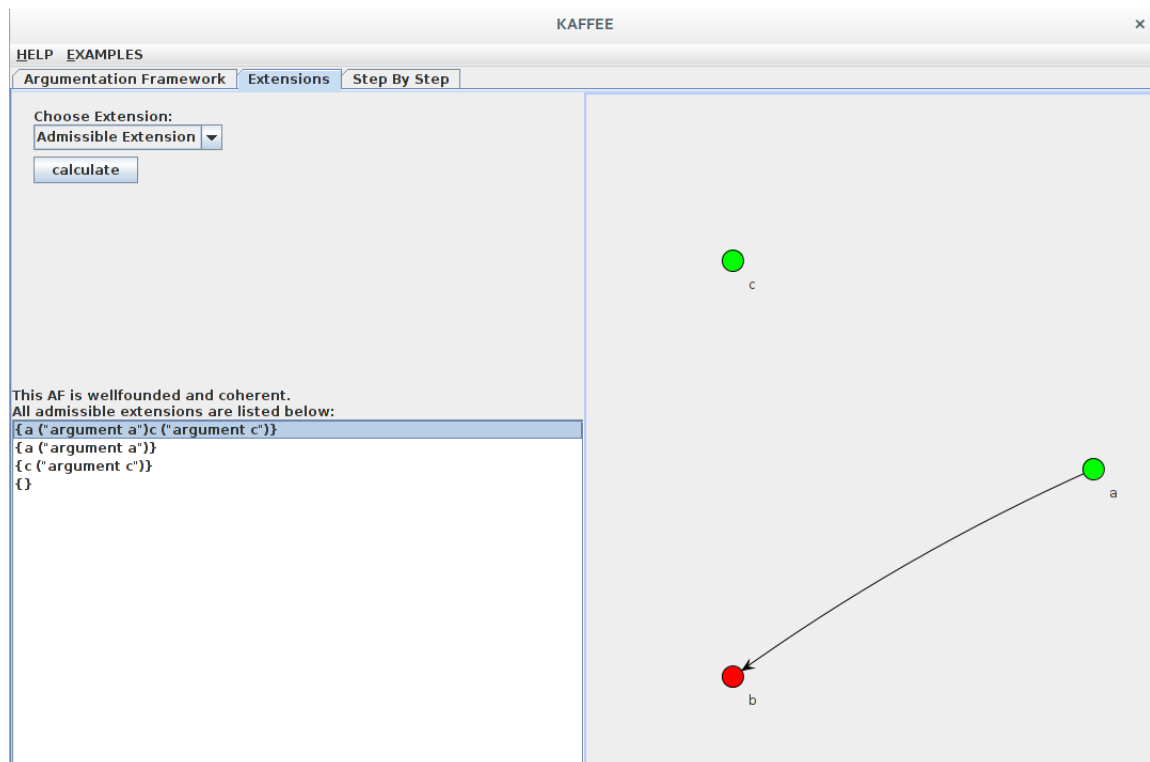


Figure 5.3: Tab "Extensions"

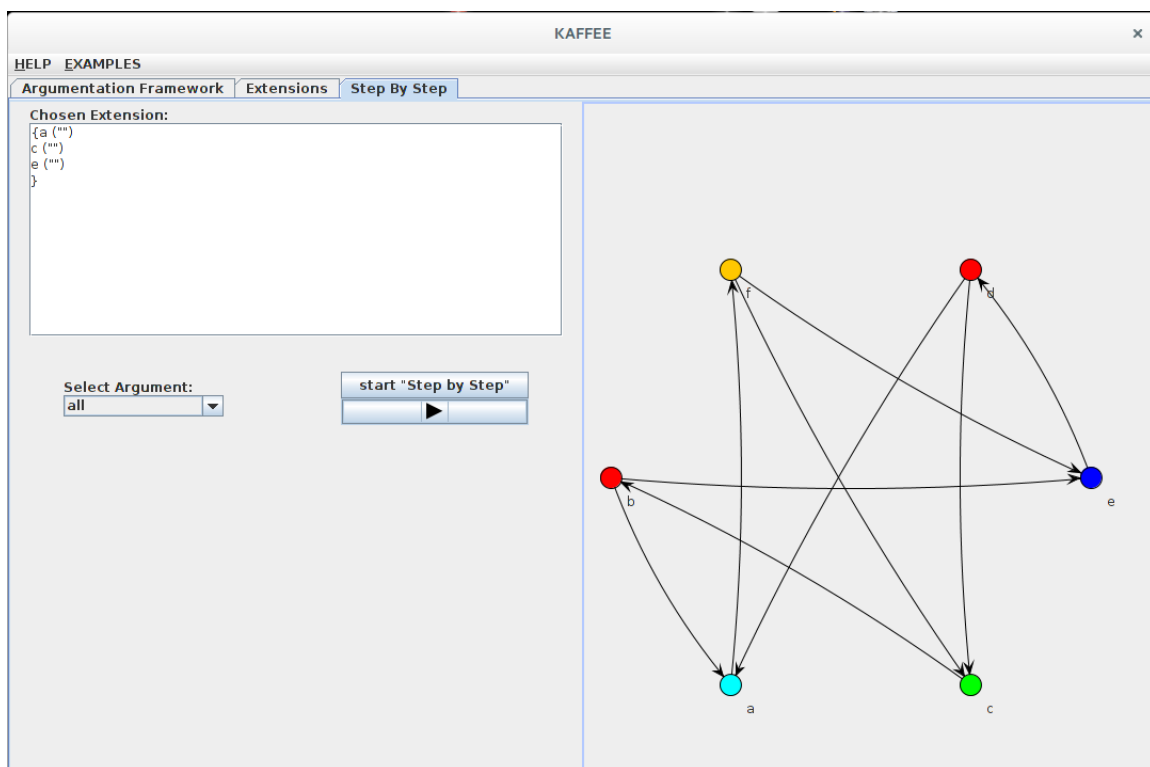


Figure 5.4: Tab "Step By Step"



# Bibliography

- [**Reiter**] "A Logic for Default Reasoning"  
Raymond Reiter; 1980
- [**Pollock**] "Defeasible Reasoning"  
John Pollock; 1987
- [**Dung**] "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning,..."  
Phan Minh Dung; 1995
- [**CaCaDu**] "Semi-stable Semantics"  
Martin Caminada, Walter Carnielli, Paul Dunne; 2006
- [**DuMaTo**] "Computing ideal sceptical argumentation"  
Phan Minh Dung, Paolo Mancarella, Francesca Toni; 2007
- [**BeHu**] "Elements of Argumentation"  
Philippe Besnard, Anthony Hunter; 2008
- [**EgGaWo**] "Answer-Set Programming Encodings for Argumentation Frameworks"  
Uwe Egly, Alice Gaggl, Stefan Woltran; 2010
- [**HuGo**] "Instantiating abstract argumentation with classical logic arguments"  
Anthony Hunter, Nikos Gorogiannis; 2011
- [**Prakken**] "Some reflections on two current trends in formal argumentation"  
Henry Prakken; 2012