FEDERAL UNIVERSITY OF CEARÁ
DEPARTMENT OF COMPUTER SCIENCE
POST-GRADUATION PROGRAM IN COMPUTER SCIENCE
MASTER DEGREE IN COMPUTER SCIENCE

RENAN DE CASTRO SILVA CORDEIRO

ON BIPOLAR ARGUMENTATION FRAMEWORKS AND THEIR EQUIVALENCE WITH LOGIC PROGRAMMING AND SETAF

FORTALEZA
2026

RENAN DE CASTRO SILVA CORDEIRO

ON BIPOLAR ARGUMENTATION FRAMEWORKS AND THEIR EQUIVALENCE WITH LOGIC PROGRAMMING AND SETAF

Dissertation submitted to the Post-Graduation Program in Computer Science of the Federal University of Ceará, as a partial requirement for obtaining the title of Master in Computer Science. Concentration Area: Theory of Computation

Advisor: Prof. Dr. João Fernando Lima Alcântara

FORTALEZA

2026

RENAN DE CASTRO SILVA CORDEIRO

ON BIPOLAR ARGUMENTATION FRAMEWORKS AND THEIR EQUIVALENCE WITH LOGIC PROGRAMMING AND SETAF

Dissertation submitted to the Post-Graduation Program in Computer Science of the Federal University of Ceará, as a partial requirement for obtaining the title of Master in Computer Science. Concentration Area: Theory of Computation

Approved on: February 23th, 2026

EXAMINATION BOARD

Prof. Dr. João Fernando Lima Alcântara (Advisor)
Universidade Federal do Ceará (UFC)

Prof. Dr. Lincoln Souza Rocha
Universidade Federal do Ceará (UFC)

Prof. Dr. Samy Soares Passos de Sá
Universidade Federal do Ceará (UFC)

Prof. Dr. Thiago Alves Rocha
Instituto Federal de Educação, Ciência e Tecnologia do Estado do Ceará (IFCE)

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

ABSTRACT

Logic programming and abstract argumentation are closely connected paradigms suitable for representing incomplete, contradictory, and/or uncertain information. In Normal Logic Programs (*NLP*s), each claim is represented by an atom, which is considered true iff it can be derived by a set of logical rules. In Abstract Argumentation Frameworks (*AAF*s), claims are encoded as arguments and evaluated solely based on an attack relation, which determines how arguments (negatively) interact with each other. Different interactions give rise to distinct argumentation variants, such as Bipolar Argumentation Frameworks (*BAF*s) and Frameworks with Sets of Attacking Arguments (*SETAF*s), that additionally allow expressing the notions of support and collective attacks, respectively. For each variant, a evaluation criteria, called *semantics*, is needed to specify precisely which claims should be regarded as *accepted*, *rejected*, or neither (*undecided*). In this work, we study *BAF*s by both a semantic and a structural perspective. We propose new semantics for *BAF*s, called β-semantics, and employ them to find semantic-preserving translations between *BAF*s, *SETAF*, and *NLP*s. Moreover, our proposed translations preserve the structure of these formalisms when restricted to redundancy-free classes of them. The resulting translations are made accessible in the following website: `https://web.archive.org/web/20260327154048/https://renpet-hhh.github.io/argnlp/static/`.

Keywords: argumentation; logic programming; intertranslatability.

RESUMO

Programação lógica e argumentação abstrata são paradigmas intimamente relacionados, adequados para representar informações incompletas, contraditórias, e/ou incertas. Em Programas Lógicos Normais (*NLP*s), cada afirmação é representada por um átomo, que é considerado verdadeiro se e somente se pode ser derivado por um conjunto de regras lógicas. Em Modelos de Argumentação Abstrata (*AAF*s), as afirmações são codificadas como argumentos e avaliadas exclusivamente com base em uma relação de ataque, que determina como os argumentos interagem (negativamente) entre si. Diferentes interações dão origem a variantes distintas de argumentação, como Modelos de Argumentação Bipolar (*BAF*s) e Modelos com Conjuntos de Argumentos Atacantes (*SETAF*s), que permitem expressar adicionalmente as noções de suporte e ataques coletivos, respectivamente. Para cada variante, um critério de avaliação, chamado *semântica*, é necessário para especificar precisamente quais afirmações devem ser consideradas *aceitas*, *rejeitadas*, ou *indecisas*. Neste trabalho, estudamos *BAF*s tanto por uma perspectiva semântica quanto estrutural. Propomos novas semânticas para *BAF*s, chamadas β-semânticas, e as empregamos para encontrar traduções entre *BAF*s, *SETAF*, e *NLP*s que preservem suas semânticas correspondentes. Além disso, as traduções propostas preservam a estrutura desses formalismos quando restritas a classes livres de redundância. As traduções resultantes estão acessíveis no seguinte site: `https://web.archive.org/web/20260327154048/https://renpet-hhh.github.io/argnlp/static/`.

Palavras-chave: argumentação; programação lógica; traduzibilidade.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Logic programming and abstract argumentation are highly successful paradigms in Knowledge Representation, specially suited for reasoning in complex domains where information is untrustful or inconsistent. Normal Logic Programs (*NLP*s) encode each claim as an atom and specify many logic-based inference rules for deriving atoms, whose evaluation consists in determining whether they can be derived from the program rules. Distinctively, argumentation-based approaches revolve around building arguments and determining how they interact. Some approaches are *structured*, in that they allow each argument to have internal structure, such as sub-arguments, evidence, and warrant. Conversely, an approach is *abstract* when each argument is evaluated as *accepted*, *rejected*, or neither (*undecided*) based solely on how they interact with each other.

In this work, we focus on abstract approaches, all of which stem from Abstract Argumentation Frameworks (*AAF*s), introduced by Dung (1995) in his seminal paper. *AAF*s are defined as a set of arguments and a binary attack relation between them. Any criteria, called *semantics*, for determining the accepted arguments must consider only how arguments attack each other. Argumentation semantics can be equivalently defined in terms of extensions (sets of accepted arguments) or labelings, which are functions mapping each argument to one of three labels: `in` (accepted), `out` (rejected), or `undec` (undecided) (Caminada, 2006; Caminada; Gabbay, 2009). We focus on labeling-based semantics as they permit a more fine-grained and explicit setting.

Despite expressing information from distinct perspectives, argumentation has been connected to logic programming since its first introduction in Dung's (1995) work, where it is already shown how to translate a *NLP* into an *AAF*. In particular, the author proved that stable models (resp. the well-founded model) of an *NLP* correspond to stable extensions (resp. the grounded extension) of the corresponding *AAF*. Since then, much effort has been made to find connections between *AAF*s and *NLP*s (Nieves; Cortés; Osorio, 2008; Wu; Caminada; Gabbay, 2009; Toni; Sergot, 2011; Dvořák; Gaggl, *et al.*, 2013; Caminada; Sá, *et al.*, 2015; Caminada; Harikrishnan; Sá, 2022). Notably, Wu; Caminada; Gabbay (2009) established the equivalence between (*AAF*) complete semantics and (*NLP*) partial stable semantics. These semantics generalize a series of other relevant semantics for each system, as extensively documented by Caminada; Sá, *et al.* (2015). However, one equivalence formerly expected to hold remained elusive: the correspondence between the semi-stable semantics (Caminada, 2006) in *AAF*s and the *L*-

stable semantics in *NLP*s (Eiter; Leone; Sacca, 1997) could not be attained. Caminada; Sá, *et al.* (2015) even showed that when using their proposed translation from *NLP*s to *AAF*s, there is no semantics for *AAF*s equivalent to the L-stable semantics. Sá; Alcântara (2021a) shed light on the reason for this mismatch and propose the (5-valued) least-stable semantics for *AAF*s, which corresponds to the *NLP* L-stable semantics.

1.1 Objectives

In our work, the general objective is to continue this line of research for more expressive argumentation formalisms. Interactions in *AAF*s are limited to attacks—intuitively, an attack from argument A to argument B represents that "when A is accepted, B is rejected"—, therefore many argumentation variants were proposed for more easily expressing other notions, such as support ("when A is accepted, so is B") and collective attacks ("when A and B are accepted, C is rejected"). In this work, we extensively discuss and compare Bipolar Argumentation Frameworks (*BAF*s) (Cayrol; Lagasquie-Schiex, 2005), which can express support explicitly, and Frameworks with Sets of Attacking Arguments (*SETAF*s) (Nielsen; Parsons, 2007), which allow expressing collective attacks. In particular, we study whether they are in semantic and syntactic correspondence with each other for a variety of semantics:

- finding semantic correspondences between classes $\mathcal{X}$ and $\mathcal{Y}$ (of formalisms) means we can translate any $X \in \mathcal{X}$ into $Y_X \in \mathcal{Y}$, and any $Y \in \mathcal{Y}$ into $X_Y \in \mathcal{X}$, such that there is a bijection between the semantics of X and Y_X, and between the semantics of Y and X_Y;
- finding syntactic correspondences between classes $\mathcal{X}$ and $\mathcal{Y}$ means we can translate any $X \in \mathcal{X}$ into $Y_X \in \mathcal{Y}$ and then into $X_{Y_X} \in \mathcal{X}$ such that X and X_{Y_X} are isomorphic, and in the other direction, from $Y \in \mathcal{Y}$ we obtain $X_Y \in \mathcal{X}$ and then $Y_{X_Y} \in \mathcal{Y}$ such that Y and Y_{X_Y} are isomorphic.

The motivation for finding such correspondences lies in the fact that they allow one to select, for each scenario, the most suitable formalism. Moreover, results, techniques, algorithms, and semantics derived from one formalism can often be adapted to the other via these transformations. It also clarifies which forms of interactions are more (or equally) expressive among those commonly used in argumentation. Besides, correspondences between *BAF*s, *SETAF*s, and *NLP*s also indirectly extend to other notable formalisms such as Abstract Dialectical Frameworks (*ADF*s) (Brewka; Woltran, 2010), Claim-Augmented Frameworks (*CAF*s) (Rapberger, 2020; Dvořák; Woltran, 2020), and Assumption-Based Argumentation (*ABA*) frameworks (Bondarenko *et al.*, 1997), by using previously known connections between *CAF*s, *ADF*s, *ABA* frameworks, and *NLP*s. Additionally, we naturally encounter and categorize many sorts of redundancies when trying to find syntactic correspondences between *BAF*s, *SETAF*s, and *NLP*s.

1.2 Contributions

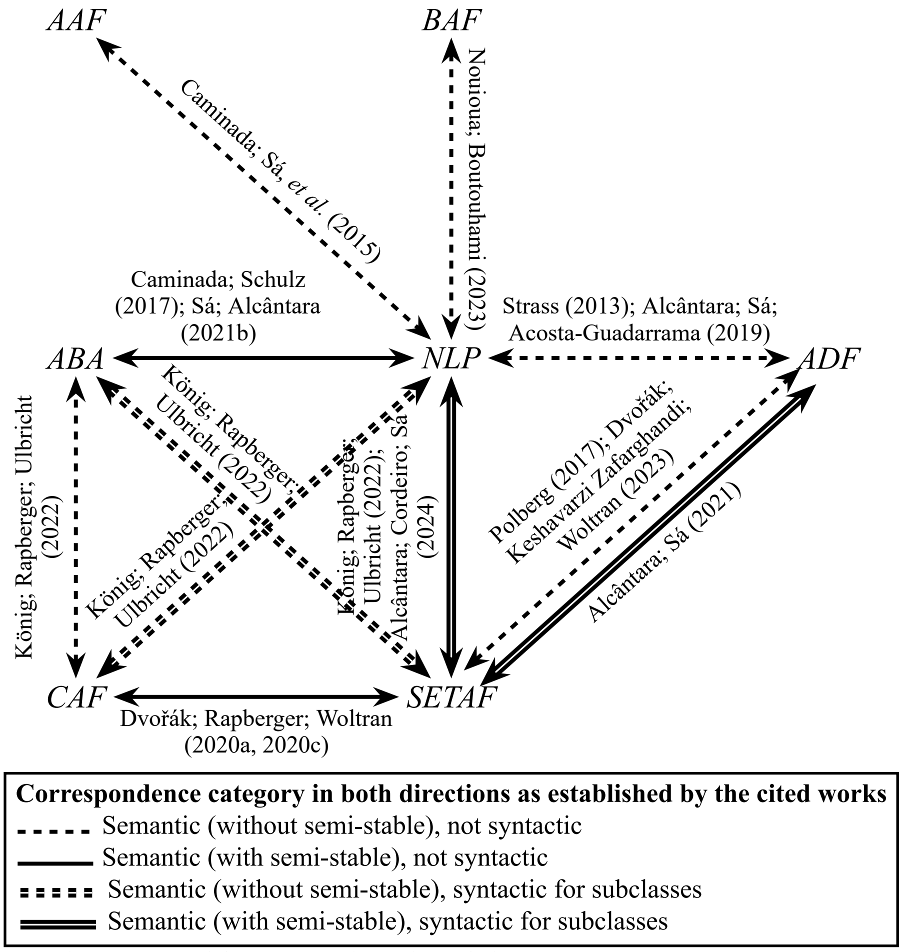


Figure 1 – Overview of notable connections involving *NLP*s, *AAF*s, *BAF*s, *SETAF*s, *ADF*s, *ABA* frameworks, and *CAF*s. The arrows indicate the kind of correspondence in two orthogonal categories: whether the cited works establish semantic correspondences including the semi-stable/L-stable semantics in *both* directions (solid lines) or not (dashed lines); and whether the cited works guarantee syntactic correspondences for *subclasses* of the formalisms (double lines) or not (single lines).

We present in Figure 1 an overview of notable connections between the formalisms mentioned so far. This dissertation contributes by deepening the connections between these formalisms, specially *BAF*s. We start by proposing a new semantics for *BAF*s and then show it corresponds to well-known semantics of *NLP*s and *SETAF*s. Our main contributions can be summarized as follows:

1. we propose new semantics for *BAF*s, called β-semantics, with a notion of support based on a dual relation between defeat and defense (Chapter 3);
2. we find one-to-one semantic correspondences between *BAF* β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable labelings and respectively *NLP* partial stable, well-founded, regular, stable, and L-stable models. Moreover, we find classes of redundancy-free *BAF*s and *NLP*s in syntactic correspondence, i.e., for which these translations are the inverse of each other up to isomorphism (Chapter 4);
3. we obtain one-to-one semantic correspondences between *BAF* β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable labelings and respectively complete, grounded, preferred, stable, and semi-stable labelings of *SETAF*s. Moreover, we find classes of redundancy-free *BAF*s and *SETAF*s in syntactic correspondence, i.e., for which these translations are the inverse of each other up to isomorphism (Chapter 5).

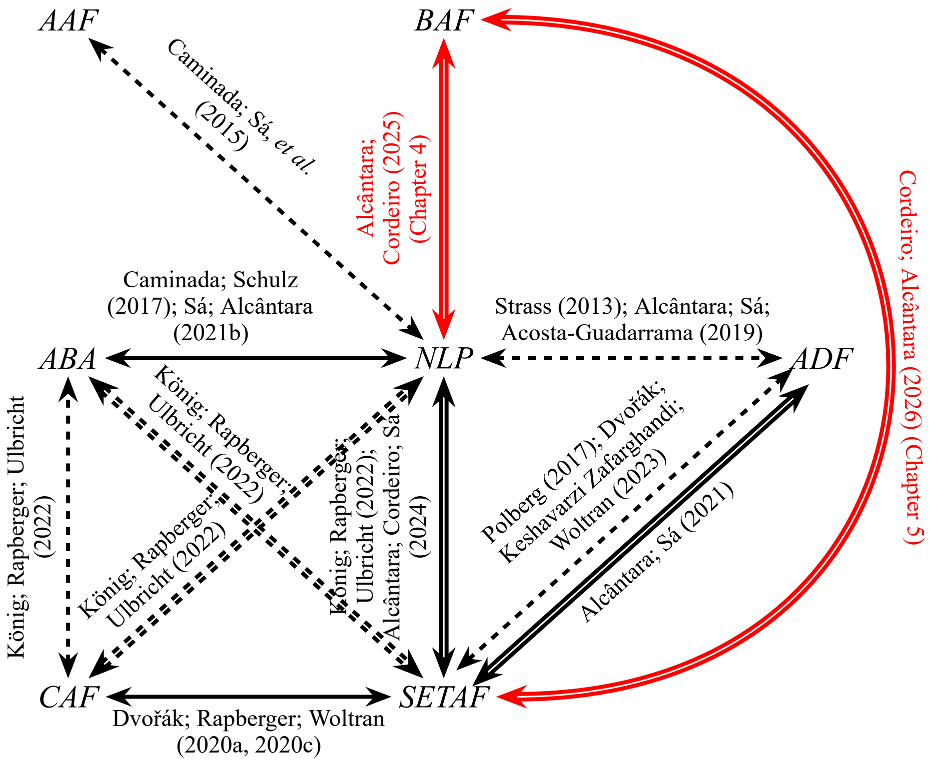


Figure 2 – How our contributions (in red) are positioned compared to the existing connections depicted in Figure 1.

Notably, the correspondences we establish above include the *NLP* L-stable semantics, which did not correspond to the *AAF* semi-stable semantics in Caminada; Sá, *et al.*'s (2015) translation between *AAF*s and *NLP*s. Moreover, these results have ramifications for other argumentation formalisms by using previously known connections between *CAF*s, *ADF*s, *ABA* frameworks, and *NLP*s. Finally, the connections we find between the β-semantics and well-established *NLP* and *SETAF* semantics also serve as additional motivation for the β-semantics (Chapter 3), showing that it provides another interesting manner of interpreting support among the many *BAF* semantics in the literature. In Figure 2, we show how our contributions are positioned compared to the existing literature.

1.3 Publications

We produced many papers as a result of this dissertation and related research:

a) ALCÂNTARA, J.; CORDEIRO, R.; SÁ, S. On the equivalence between logic programming and SETAF. **Theory and Practice of Logic Programming**, [Cambridge], v. 24, n. 6, p. 1208–1236, 2024. DOI: `10.1017/S1471068424000188`;
b) ALCÂNTARA, J.; CORDEIRO, R. Bipolar Argumentation Frameworks with a dual relation between defeat and defence. **Journal of Logic and Computation**, [Oxford], v. 35, n. 2, p. exae006, 2024. DOI: `10.1093/logcom/exae006`;
c) ALCÂNTARA, J.; CORDEIRO, R. On the Equivalence between Logic Programs and Bipolar Argumentation Frameworks. **Journal of Artificial Intelligence Research**, [El Segundo], v. 84, n. 17, p. 1–43, 2025. DOI: `10.1613/jair.1.18086`;
d) CORDEIRO, R.; ALCÂNTARA, J. On the Equivalence between Bipolar Argumentation Frameworks and SETAFs. **Argument & Computation**, [Amsterdam], 2026. In press.

In these works, we show the connection between *NLP*s and *SETAF*s (Alcântara; Cordeiro; Sá, 2024); a new β-semantics for *BAF*s (Alcântara; Cordeiro, 2024); the connection between *BAF*s and *NLP*s (Alcântara; Cordeiro, 2025); and the connection between *BAF*s and *SETAF*s (Cordeiro; Alcântara, 2026). Additionally, a paper about the connection between *NLP*s and Argumentation Frameworks with Recursive Attacks (*AFRA*s) is being prepared for submission.

1.4 Overview

Before proceeding to the contributions, we briefly review the necessary background in Chapter 2, such as the definitions of *AAF*s, *BAF*s, *SETAF*s, *NLP*s, and their corresponding semantics considered in our work. In Chapter 3, we present our new β-semantics for *BAF*s. Then, we show that *BAF*s under the β-semantics are deeply connected to *NLP*s in Chapter 4 and to *SETAF*s in Chapter 5. Finally, we conclude in Chapter 6 by summarizing our contributions, highlighting limitations, and discussing future work.

2 PRELIMINARIES

This chapter introduces the formalisms and semantics underlying our work. We start by reviewing some mathematical concepts employed thoroughout the dissertation. Next, we proceed to Abstract Argumentation Frameworks (AAFs), which lie at the core of all argumentation formalisms studied in our work, from which follows Bipolar Argumentation Frameworks (BAFs) and Frameworks with Sets of Attacking Arguments (SETAFs). Then, we move on to Normal Logic Programs (NLPs) and their 3-valued semantics. Lastly, we present Caminada; Sá, et al.*'s (2015) correspondence results between AAFs and NLPs, which our approaches are based upon.*

2.1 Mathematical background

A *partially ordered set* (poset) is a pair $(S, \preceq)$ such that S is a set and $\preceq$ is a binary relation on S satisfying three properties:

a) **(reflexivity)** for every $x \in S$, it holds $x \preceq x$;

b) **(antisymmetry)** for every $x, y \in S$, if $x \preceq y$ and $y \preceq x$, then $x = y$;

c) **(transitivity)** for every $x, y, z \in S$, if $x \preceq y$ and $y \preceq z$, then $x \preceq z$.

A *sub-poset* of $(S, \preceq)$ is a poset $(S', \preceq')$ where $S' \subseteq S$ and $\preceq' = (S' \times S') \cap \preceq$. A poset $(S, \preceq)$ is a *totally ordered set* iff for all $x, y \in S$, either $x \preceq y$ or $y \preceq x$. A *chain* in $(S, \preceq)$ is a sub-poset of $(S, \preceq)$ that is a totally ordered set.

For a poset $(S, \preceq)$, we say $x \in S$ is an *upper bound* for $S' \subseteq S$ iff $(\forall y \in S')y \preceq x$. Similarly, $x \in S$ is a *lower bound* for S' iff $(\forall y \in S')x \preceq y$. The least upper bound (supremum) of S', denoted by *lub*(S'), is an *upper bound* for S' such that for any *upper bound* y for S', it holds *lub*$(S') \preceq y$. In addition, the greatest lower bound (infimum) of S', denoted by *glb*(S'), is an *lower bound* for S' such that for any *lower bound* y for S', it holds $y \preceq$ *glb*(S'). We emphasize that depending on S', *glb*(S') and *lub*(S') may not exist, but if each does exist, then each is unique.

Let $(S, \preceq)$ be a poset. A function $f : S \to S$ is *monotonic* iff $(\forall x, y \in S)$ if $x \preceq y$, then $f(x) \preceq f(y)$. We say $(S, \preceq)$ is *directed* iff every pair of (not necessarily distinct) elements in S has an upper bound in S. A poset $(S, \preceq)$ is *directed-complete* (dcpo) or simply complete iff every directed subset has its least upper bound in S. Furthermore, $(S, \preceq)$ is a complete semilattice iff each nonempty subset of S has a greatest lower bound and, for each chain $(S', \preceq')$ of $(S, \preceq)$, the set S' has a least upper bound.

We recall useful definitions from the combinatorics of finite sets (Berge, 1984).

Definition 1 (Hypergraph, Berge 1984)**.** Let $\mathcal{X}$ be a finite set. A hypergraph on $\mathcal{X}$ is a family $\mathbf{H} \subseteq 2^{\mathcal{X}}$ of subsets of $\mathcal{X}$ such that (i) $\mathcal{H} \neq \emptyset$ for any $\mathcal{H} \in \mathbf{H}$, and (ii) $\bigcup_{\mathcal{H} \in \mathbf{H}} \mathcal{H} = \mathcal{X}$.

Hypergraphs with only $\subseteq$-minimal elements are called *simple*.

Definition 2 (Simple hypergraph, Berge 1984)**.** A hypergraph $\mathbf{H}$ on $\mathcal{X}$ is said to be *simple* when there are only $\subseteq$-minimal elements in $\mathbf{H}$, i.e., there is no $\mathcal{H}, \mathcal{H}' \in \mathbf{H}$ such that $\mathcal{H} \subset \mathcal{H}'$.

A transversal is a subset of $\mathcal{X}$ intersecting each element of $\mathbf{H}$, as described next:

Definition 3 (Transversal, Berge 1984)**.** Let $\mathbf{H}$ be a hypergraph on $\mathcal{X}$. A set $\mathcal{T} \subseteq \mathcal{X}$ is a transversal of $\mathbf{H}$ if $\mathcal{T}$ intersects each element of $\mathbf{H}$, i.e., $\mathcal{T} \cap \mathcal{H} \neq \emptyset$ for any $\mathcal{H} \in \mathbf{H}$. A transversal is also called a hitting set in other works.

The family of $\subseteq$-minimal transversals of a hypergraph $\mathbf{H}$ on $\mathcal{X}$ is denoted by $\mathrm{Tr}(\mathbf{H})$ and is always a (simple) hypergraph.

Definition 4 ($\mathrm{Tr}(\cdot)$ operator, Berge 1984)**.** Let $\mathbf{H}$ be a hypergraph on $\mathcal{X}$. The family of $\subseteq$-minimal transversals of $\mathbf{H}$ is denoted by $\mathrm{Tr}(\mathbf{H})$.

Berge (1984) proved that the Tr operator is involutive for the class of simple hypergraphs. Involutive means that applying Tr twice restores the original family $\mathbf{H}$, i.e., $\mathrm{Tr}(\mathrm{Tr}(\mathbf{H})) = \mathbf{H}$ for any simple hypergraph $\mathbf{H}$.

Theorem 1 (Berge 1984)**.** Let $\mathbf{H}$ be a simple hypergraph on $\mathcal{X}$. Then $\mathrm{Tr}(\mathrm{Tr}(\mathbf{H})) = \mathbf{H}$.

Below, we contextualize these results to argumentation. First, we extend the definition of transversals so that they can be applied to any family $\mathbf{H}$ of subsets of $\mathcal{X}$ and not just to hypergraphs on $\mathcal{X}$. This is useful because the properties of hypergraphs (Definition 1) are not relevant to our work.

Definition 5 (Transversal in argumentation)**.** Let $\mathcal{A}$ be a finite set of arguments. Let $\mathbf{H} \subseteq 2^{\mathcal{A}}$ be a family of subsets of $\mathcal{A}$. A set $\mathcal{T} \subseteq \mathcal{A}$ is a transversal of $\mathbf{H}$ if $\mathcal{T}$ intersects each element of $\mathbf{H}$, i.e., $\mathcal{T} \cap \mathcal{H} \neq \emptyset$ for any $\mathcal{H} \in \mathbf{H}$. A transversal is also called a hitting set in other works.

Both notions of transversals in Definitions 3 and 5 are the same. The only difference is whether $\mathbf{H}$ is required to be a hypergraph. To be absolutely clear, we present the definition of the Tr operator for families of subsets of $\mathcal{A}$:

Definition 6 (Tr operator)**.** Let $\mathbf{H} \subseteq 2^{\mathcal{A}}$ be a family of subsets of $\mathcal{A}$. The family of $\subseteq$-minimal transversals of $\mathbf{H}$ (as in Definition 5) is denoted by $\mathrm{Tr}[\mathbf{H}]$.

We use brackets $[\cdot]$ instead of parenthesis $(\cdot)$ to distinguish from the Tr operator in Definition 4, but these operators, when defined, coincide. The only situation in which we cannot affirm $\mathrm{Tr}(\mathbf{H}) = \mathrm{Tr}[\mathbf{H}]$ is when $\mathrm{Tr}(\mathbf{H})$ is not defined (because $\mathbf{H}$ is not a hypergraph).

As we show next, we can still obtain the involutive property of Tr, independent of whether $\mathbf{H}$ is a hypergraph:

Theorem 2. Let $\mathcal{A}$ be a finite set of arguments. Let $\mathbf{H} \subseteq 2^{\mathcal{A}}$ be a family of subsets of $\mathcal{A}$ where there is no $\mathcal{H}, \mathcal{H}' \in \mathbf{H}$ such that $\mathcal{H} \subsetneq \mathcal{H}'$. Then $\mathrm{Tr}[\mathrm{Tr}[\mathbf{H}]] = \mathbf{H}$.

As the Tr operator will be used in the translations between *BAF*s and *SETAF*s (defined in the next sections), Theorem 2 is fundamental in establishing syntactic correspondences between subclasses of *BAF*s and *SETAF*s.

2.2 Abstract Argumentation Frameworks (*AAF*s)

Abstract Argumentation Frameworks (*AAF*s) are defined by Dung (1995) as a set of arguments and an attack relation between arguments. *AAF*s can be represented by directed graphs where nodes represent arguments and arrows correspond to attacks.

Definition 7 (Abstract Argumentation Framework, *AAF*, Dung 1995)**.** An Abstract Argumentation Framework (*AAF*) is a pair $\mathfrak{F} = (\mathcal{A}, \mathit{Att})$, where $\mathcal{A}$ is a set of arguments and $\mathit{Att} \subseteq \mathcal{A} \times \mathcal{A}$ is the attack relation between arguments. When $(A, B) \in \mathit{Att}$, we say A *attacks* B. For any $A \in \mathcal{A}$, we say $\mathit{Att}(A) = \{B \in \mathcal{A} \mid (B, A) \in \mathit{Att}\}$ is the set of *attackers* of A.

Criteria for evaluating arguments are called *semantics*, and they were initially characterized by Dung (1995) in terms of sets of arguments, called *extensions*. As arguments are viewed as abstract entities whose internal structure is irrelevant, *AAF* semantics take into account solely how an argument interacts in the framework. *AAF* semantics revolve around the idea of sets of arguments holding a stance. To be reasonable, a stance must not admit contradictions.

This is formalized by the concept of conflict-free sets. Throughout this section, let $\mathfrak{F} = (\mathcal{A}, Att)$ be an *AAF*.

Definition 8 (Conflict-free set, Dung 1995)**.** Let $\mathcal{S} \subseteq \mathcal{A}$ be a set of arguments. We say $\mathcal{S}$ is *conflict-free* in $\mathfrak{F}$ iff for every arguments $A, B \in \mathcal{A}$, it holds $(A, B) \notin Att$.

Example 1. Let $\mathfrak{F} = (\mathcal{A}, Att)$ be the *AAF* such that $\mathcal{A} = \{A, B, C, D, E, F\}$ and $Att = \{(A, B), (B, A), (B, C), (C, F), (D, D), (E, C), (F, E)\}$, depicted in Figure 3. Recall that we use nodes to represent arguments and arrows to represent attacks. The sets $\{A, C\}$ and $\{B, F\}$ are conflict-free in the *AAF* from Figure 3, but $\{A, B\}$, $\{D\}$, and $\{B, C, F\}$ are not.

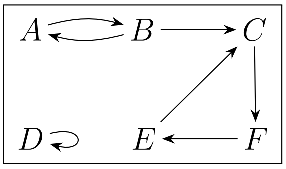


Figure 3 – An Abstract Argumentation Framework (*AAF*).

When obvious, we simply say $\mathcal{S}$ is conflict-free without mentioning $\mathfrak{F}$ (and similarly for the definitions below). Intuitively, in a conflict-free set, the attack (A, B) means that if A is accepted, then B is rejected. In addition to being conflict-free, a reasonable stance must be capable of defending itself against incoming attackers. For that, we introduce the concepts of *defeat* and *defense* in *AAF*s:

Definition 9 (Defeat, Dung 1995)**.** Let $A \in \mathcal{A}$ be an argument and $\mathcal{S} \subseteq \mathcal{A}$ a set of arguments. We say $\mathcal{S}$ *defeats* A in $\mathfrak{F}$ iff there exists $B \in \mathcal{S}$ such that $(B, A) \in Att$.

We denote by $\mathcal{S}^+$ the set $\{A \in \mathcal{A} \mid B \in \mathcal{S} \textit{ and } (B, A) \in Att\}$ of arguments defeated by $\mathcal{S}$. From the definition of defeat follows that of defense:

Definition 10 (Defense, Dung 1995)**.** Let $A \in \mathcal{A}$ be an argument and $\mathcal{S} \subseteq \mathcal{A}$ a set of arguments. We say $\mathcal{S}$ *defends* A in $\mathfrak{F}$ iff for every $B \in Att(A)$, it holds $\mathcal{S}$ defeats B in $\mathfrak{F}$.

Instead of saying that $\mathcal{S}$ defends A, some authors say that A is acceptable w.r.t. $\mathcal{S}$. We denote by $F_{\mathfrak{F}}(\mathcal{S})$ the set of arguments defended by $\mathcal{S}$ in $\mathfrak{F}$. The operator $F_{\mathfrak{F}}$ is given the special name of *characteristic function*.

Definition 11 (Characteristic function, Dung 1995)**.** The characteristic function $F_{\mathfrak{F}} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ is defined as

$$F_{\mathfrak{F}}(\mathcal{S}) = \{A \in \mathcal{A} \mid A \textit{ is defended by } \mathcal{S}\}.$$

A conflict-free set $\mathcal{S}$ such that $\mathcal{S} \subseteq F_{\mathfrak{F}}(\mathcal{S})$ is called *admissible*. It represents a stance without contradictions and capable of defeating each of its attackers. This notion of admissibility is fundamental to all *AAF* semantics we consider in this paper. The original definitions from Dung (1995) are equivalently presented below following the (simplified) outline given by Caminada; Gabbay (2009).

Definition 12 (Extension-based semantics for *AAF*s, Dung 1995; Caminada; Gabbay 2009)**.** Let $\mathcal{S} \subseteq \mathcal{A}$ be a set of arguments in *AAF* $\mathfrak{F} = (\mathcal{A}, \mathit{Att})$:

a) $\mathcal{S}$ is *complete* iff $\mathcal{S}$ is a conflict-free fixpoint of $F_{\mathfrak{F}}$, i.e., $\mathcal{S}$ is conflict-free and $\mathcal{S} = F_{\mathfrak{F}}(\mathcal{S})$;
b) $\mathcal{S}$ is *grounded* iff $\mathcal{S}$ is a $\subseteq$-minimal conflict-free fixpoint of $F_{\mathfrak{F}}$;
c) $\mathcal{S}$ is *preferred* iff $\mathcal{S}$ is a $\subseteq$-maximal conflict-free fixpoint of $F_{\mathfrak{F}}$;
d) $\mathcal{S}$ is *stable* iff $\mathcal{S}$ is a conflict-free fixpoint of $F_{\mathfrak{F}}$ such that $\mathcal{S} \cup \mathcal{S}^+ = \mathcal{A}$;
e) $\mathcal{S}$ is *semi-stable* iff $\mathcal{S}$ is a conflict-free fixpoint of $F_{\mathfrak{F}}$ with $\subseteq$-maximal $\mathcal{S} \cup \mathcal{S}^+$.

A conflict-free fixpoint $\mathcal{S}$ of $F_{\mathfrak{F}}$ (a complete extension $\mathcal{S}$) has no conflicts, defends each argument in $\mathcal{S}$ (i.e., $\mathcal{S} \subseteq F_{\mathfrak{F}}(\mathcal{S})$), and contains every argument it defends (i.e., $F_{\mathfrak{F}}(\mathcal{S}) \subseteq \mathcal{S}$). Complete extensions represent the reasonable stances one could take when considering only the information about how arguments interact with each other. Additionally, some optimization criteria can be used to refine a stance, such as minimizing/maximizing the set of accepted arguments (grounded/preferred extensions), or minimizing the set of undecided arguments (semi-stable extensions).

Example 2. Recall the *AAF* $\mathfrak{F}$ from Example 1. It has the following extension-based semantics:

a) complete extensions $\emptyset$, $\{A\}$, and $\{B, F\}$;
b) grounded extension $\emptyset$;
c) preferred extension $\{A\}$ and $\{B, F\}$;
d) no stable extensions;
e) semi-stable extension $\{B, F\}$.

The extension-based semantics above can be equivalently be expressed as labeling-based semantics. In contrast to extensions, labelings allow for a more fine-grained setting in which each argument is associated with a explicit acceptability label among the labels `in`, `out`, and `undec`. Intuitively, an argument labeled `in` is accepted; an argument labeled `out` is rejected; and one labeled `undec` is undecided, i.e., neither accepted nor rejected.

Definition 13 (*AAF* Labeling)**.** A *labeling* of *AAF* $\mathfrak{F} = (\mathcal{A}, Att)$ is a total function $\mathcal{L} : \mathcal{A} \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$.

We present the labeling-based semantics according to the outline provided by Caminada; Gabbay (2009):

Definition 14 (Labeling-based semantics for *AAF*s, Caminada; Gabbay 2009)**.** A labeling $\mathcal{L}$ of *AAF* $\mathfrak{F} = (\mathcal{A}, Att)$ is *complete* iff for any $A \in \mathcal{A}$, the following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(B) = \texttt{out}$ for every $B \in Att(A)$;
b) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(B) = \texttt{in}$ for some $B \in Att(A)$.

Let $\mathcal{L}$ be a complete labeling of $\mathfrak{F}$:

a) $\mathcal{L}$ is *grounded* if $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among the complete labelings of $\mathfrak{F}$;
b) $\mathcal{L}$ is *preferred* if $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal among the complete labelings of $\mathfrak{F}$;
c) $\mathcal{L}$ is *stable* if $\texttt{undec}(\mathcal{L}) = \emptyset$;
d) $\mathcal{L}$ is *semi-stable* if $\texttt{undec}(\mathcal{L})$ is $\subseteq$-minimal among the complete labelings of $\mathfrak{F}$.

We write $\texttt{in}(\mathcal{L})$ to denote the set $\{A \in \mathcal{A} \mid \mathcal{L}(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L}) = \{A \in \mathcal{A} \mid \mathcal{L}(A) = \texttt{out}\}$, and $\texttt{undec}(\mathcal{L}) = \{A \in \mathcal{A} \mid \mathcal{L}(A) = \texttt{undec}\}$. As a labeling defines a partition of $\mathcal{A}$, when convenient we write $\mathcal{L}$ as a triple $(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ where $\mathcal{S}_1 = \texttt{in}(\mathcal{L})$, $\mathcal{S}_2 = \texttt{out}(\mathcal{L})$, and $\mathcal{S}_3 = \texttt{undec}(\mathcal{L})$.

Example 3. Recall the *AAF* $\mathfrak{F}$ from Example 1. It has the following labeling-based semantics:

a) complete labelings $\mathcal{L}_1 = (\emptyset, \emptyset, \mathcal{A})$, $\mathcal{L}_2 = (\{A\}, \{B\}, \{C, D, E, F\})$, and $\mathcal{L}_3 = (\{B, F\}, \{A, C, E\}, \{D\})$;
b) grounded labeling $\mathcal{L}_1$;
c) preferred labelings $\mathcal{L}_2$ and $\mathcal{L}_3$;
d) no stable labelings;
e) semi-stable labeling $\mathcal{L}_3$.

Caminada; Gabbay (2009) shows there is a one-to-one correspondence between complete, grounded, preferred, stable, and semi-stable extensions and respectively complete, grounded, preferred, stable, and semi-stable labelings for *AAF*s. Each extension $\mathcal{S}$ correspond to the labeling $\mathcal{L}$ with $\texttt{in}(\mathcal{L}) = S$ and $\texttt{out}(\mathcal{L}) = S^+$. Conversely, each labeling $\mathcal{L}$ corresponds to the extension $\texttt{in}(\mathcal{L})$. Note that the arguments labeled $\texttt{out}$ are precisely the arguments defeated by the set of arguments labeled $\texttt{in}$. Hence, these semantics for *AAF*s can be defined in terms of extensions or labelings in an equivalent way. We will focus on labeling-based semantics in this dissertation as they inform both the acceptance and rejection of arguments explicitly.

Next, we review frameworks with both an attack and a support relation between arguments.

2.3 Bipolar Argumentation Frameworks (*BAF*s)

Bipolar Argumentation Frameworks (*BAF*s) (Cayrol; Lagasquie-Schiex, 2005) have emerged as an extension of *AAF*s by incorporating an explicit support relation between arguments. *BAF*s can be represented by directed graphs where nodes represent arguments, solid arrows represent attacks, and dashed arrows represent the support relation.

Definition 15 (Bipolar Argumentation Framework, *BAF*, Cayrol; Lagasquie-Schiex 2005)**.** A Bipolar Argumentation Frameworks (*BAF*) is a tuple $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$, where $\mathcal{A}$ is a set of arguments, $\mathit{Att} \subseteq \mathcal{A} \times \mathcal{A}$ is the attack relation, and $\mathit{Sup} \subseteq \mathcal{A} \times \mathcal{A}$ is the support relation. When $(A, B) \in \mathit{Att}$, we say A *attacks* B. When $(A, B) \in \mathit{Sup}$, we say A *supports* B. For any $A \in \mathcal{A}$, we say $\mathit{Att}(A) = \{B \in \mathcal{A} \mid (B, A) \in \mathit{Att}\}$ is the set of *attackers* of A. For any $A \in \mathcal{A}$, we say $\mathit{Sup}(A) = \{B \in \mathcal{A} \mid (B, A) \in \mathit{Sup}\}$ is the set of *direct supporters* of A.

Compared to *AAF*s, the novelty in *BAF*s is that arguments can not only attack but also support arguments, so that *BAF*s $(\mathcal{A}, \mathit{Att}, \mathit{Sup})$ with $\mathit{Sup} = \emptyset$ amount to standard *AAF*s. This enrichment enhances the expressive power of *AAF*s, allowing for a more nuanced representation of the interactions between arguments. However, one additional challenge appears in *BAF*s: the lack of consensus in interpreting the support relation and their combination with the attack relation. This ambiguity has led to diverse approaches and varying semantics for *BAF*s (Amgoud *et al.*, 2008; Oren; Norman, 2008; Boella *et al.*, 2010; Nouioua; Risch, 2011; Cayrol; Lagasquie-Schiex, 2013; Cohen *et al.*, 2014).

In this dissertation, we propose yet a new semantics for *BAF*s by leveraging a dual relation between the two essential notions for argumentation frameworks: defeat and defense. We resort to discussing the many existing *BAF* semantics only in Chapter 3 where the new semantics is presented. We intend to show that this semantics provides a useful (yet unexplored) perspective to the interaction between attacks and supports. Remarkably, we observe that it will permit relating *BAF*s to other formalisms studied in our work, such as *SETAF*s and *NLP*s. This includes the L-stable semantics of *NLP*s, whose correspondence to semi-stable semantics in *AAF*s is not guaranteed by Caminada; Sá, *et al.*'s (2015) approach.

Now, we proceed to frameworks with collective attacks.

2.4 Frameworks with Sets of Attacking Arguments (*SETAF*s)

Frameworks with Sets of Attacking Arguments (*SETAF*s) (Nielsen; Parsons, 2007) extend the attack relation in *AAF*s to allow the representation of *collective attacks* (also called *joint attacks*), which originate from a set of arguments.

Definition 16 (Frameworks with Sets of Attacking Arguments, *SETAF*s, Nielsen; Parsons 2007)**.** A Framework with Sets of Attacking Arguments (*SETAF*) is a pair $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$, in which $\mathcal{A}$ is a finite set of arguments and $\mathit{Att} \subseteq (2^{\mathcal{A}} - \{\emptyset\}) \times \mathcal{A}$ is an attack relation such that if $(\mathcal{X}, A) \in \mathit{Att}$, then there is no $\mathcal{X}' \subset \mathcal{X}$ such that $(\mathcal{X}', A)$. When $(\mathcal{X}, A)$, we say $\mathcal{X}$ attacks A. For any $A \in \mathcal{A}$, we say $\mathit{Att}(A) = \{\mathcal{X} \subseteq \mathcal{A} \mid (\mathcal{X}, A) \in \mathit{Att}\}$ is the set of attackers of A.

Compared to *AAF*s, the attack relation *Att* is modified to relate sets of arguments to arguments. Observe that *SETAF*s $(\mathcal{A}, \mathit{Att})$ with $|\mathcal{X}| = 1$ for each $\mathcal{X} \in \mathit{Att}(A)$ and $A \in \mathcal{A}$ amount to standard *AAF*s. Definition 16 includes a minimality constraint: if $\mathcal{X} \in \mathit{Att}(A)$, then $\mathcal{X}$ is a $\subseteq$-minimal set attacking A. This condition is not present in Nielsen; Parsons's (2007) original definition of *SETAF*s, but Polberg (2017, p. 135) and Dvořák; Rapberger; Woltran (2020b) prove that this restriction can be added without changing the semantics considered in our work.

Similar to *AAF*s, semantics for *SETAF*s can be characterized in terms of extensions or labelings. We will focus on labelings. Thoroughout this section, let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF*.

Definition 17 (*SETAF* Labeling)**.** A labeling of *SETAF* $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ is a total function $\mathcal{L} : \mathcal{A} \to \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$.

We reuse the same convention for *AAF* labelings: $\mathtt{in}(\mathcal{L}) = \{A \in \mathcal{A} \mid \mathcal{L}(A) = \mathtt{in}\}$, $\mathtt{out}(\mathcal{L}) = \{A \in \mathcal{A} \mid \mathcal{L}(A) = \mathtt{out}\}$, and $\mathtt{undec}(\mathcal{L}) = \{A \in \mathcal{A} \mid \mathcal{L}(A) = \mathtt{undec}\}$. When convenient, we write $\mathcal{L}$ as the triple $(\mathtt{in}(\mathcal{L}), \mathtt{out}(\mathcal{L}), \mathtt{undec}(\mathcal{L}))$.

Intuitively, a collective attack from $\mathcal{X} \subseteq \mathcal{A}$ to $A \in \mathcal{A}$ means that if *every* argument in $\mathcal{X}$ is accepted, then A is rejected. Conversely, if *some* argument in $\mathcal{X}$ is rejected, then the entire attack is ineffective and can be disregarded. We use Flouris; Bikakis's (2019) labeling-based characterization of Nielsen; Parsons's (2007) extension-based semantics for *SETAF*s:

Definition 18 (Labeling-based semantics for *SETAF*s, Flouris; Bikakis 2019)**.** A labeling $\mathcal{L}$ of *SETAF* $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ is *complete* iff for any $A \in \mathcal{A}$, the following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(\mathcal{X}) = \texttt{out}$ for every $\mathcal{X} \in \mathit{Att}(A)$, there exists $X \in \mathcal{X}$ such that $\mathcal{L}(X) = \texttt{out}$;

b) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(\mathcal{X}) = \texttt{in}$ for some $\mathcal{X} \in \mathit{Att}(A)$, there exists $X \in \mathcal{X}$ such that $\mathcal{L}(X) = \texttt{in}$.

Let $\mathcal{L}$ be a complete labeling of $\mathfrak{A}$:

a) $\mathcal{L}$ is *grounded* if $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among the complete labelings of $\mathfrak{A}$;

b) $\mathcal{L}$ is *preferred* if $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal among the complete labelings of $\mathfrak{A}$;

c) $\mathcal{L}$ is *stable* if $\texttt{undec}(\mathcal{L}) = \emptyset$;

d) $\mathcal{L}$ is *semi-stable* if $\texttt{undec}(\mathcal{L})$ is $\subseteq$-minimal among the complete labelings of $\mathfrak{A}$.

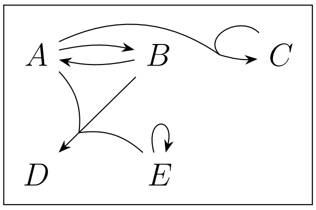


Figure 4 – *SETAF* $\mathfrak{A}$ from Example 4.

Example 4. Consider the *SETAF* $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ depicted in Figure 4, where nodes represent arguments, and arrows (with one head and possibly multiple tails) represent the attack relation. For instance, the arrow starting at A, B, E, and ending at D indicates that $\{A, B, E\}$ is an attacker of D. The *SETAF* $\mathfrak{A}$ has the following labeling-based semantics:

a) complete labelings $\mathcal{L}_1 = (\emptyset, \emptyset, \{A, B, C, D, E\})$, $\mathcal{L}_2 = (\{A, D\}, \{B\}, \{C, E\})$, and $\mathcal{L}_3 = (\{B, C, D\}, \{A\}, \{E\})$;

b) grounded labeling $\mathcal{L}_1$;

c) preferred labelings $\mathcal{L}_2$ and $\mathcal{L}_3$;

d) no stable labelings;

e) semi-stable labeling $\mathcal{L}_3$.

In the next section, we move on from argumentation formalisms to logic programming.

2.5 Normal Logic Programs (*NLP*s)

Normal Logic Programs (*NLP*s) (Przymusinski, 1990) are sets of logical rules that describe how atoms (representing claims) can be derived. The definition below follows the presentation outlined by Caminada; Sá, *et al.* (2015):

Definition 19 (Normal Logic Program, *NLP*, Caminada; Sá, *et al.* 2015)**.** A *rule* r is an expression

$$r : c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$$

where $n, m \geq 0$, each a_i $(1 \leq i \leq n)$, each b_j $(1 \leq j \leq m)$ and c are atoms, and `not` represents negation as failure. A literal is either an atom (positive literal) or a negated atom $\texttt{not}\ b$ (negative literal). Given a rule r as above, we say $\mathit{head}(r) = c$ is the *head* of r, and $\mathit{body}(r) = \{a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$ is the *body* of r. Further, we partition $\mathit{body}(r)$ into two sets $\mathit{body}^+(r) = \{a_1, \ldots, a_n\}$ and $\mathit{body}^-(r) = \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$. A *fact* is a rule with empty body ($n = m = 0$). A Normal Logic Program (*NLP*) P is a finite set of rules. If $\mathit{body}^-(r) = \emptyset$ for every $r \in P$, we say P is a positive program. The *Herbrand Base* of P is the set HB_P of all atoms appearing in P.

The semantics considered in our work are based on the 3-valued interpretations of *NLP*s (Przymusinski, 1990):

Definition 20 (3-Valued Interpretation, Przymusinski 1990)**.** A 3-valued Herbrand Interpretation $\mathcal{I}$ (or simply interpretation) of an *NLP* P is a total function $\mathcal{I} : \mathit{HB}_P \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$.

Given an interpretation $\mathcal{I}$ of an *NLP* P, we write $\mathbf{t}(\mathcal{I})$ to denote the set $\{a \in \mathit{HB}_P \mid \mathcal{I}(a) = \mathbf{t}\}$, $\mathbf{f}(\mathcal{I})$ for $\{a \in \mathit{HB}_P \mid \mathcal{I}(a) = \mathbf{f}\}$, and $\mathbf{u}(\mathcal{I})$ for $\{a \in \mathit{HB}_P \mid \mathcal{I}(a) = \mathbf{u}\}$. When convenient, we write $\mathcal{I}$ as the triple $(\mathbf{t}(\mathcal{I}), \mathbf{f}(\mathcal{I}), \mathbf{u}(\mathcal{I}))$. Intuitively, the value **t** indicates the atom is true in $\mathcal{I}$, the value **f** indicates the atom is false in $\mathcal{I}$, and the value **u** indicates the atom is undefined in $\mathcal{I}$, i.e., neither true nor false.

The main semantics for *NLP*s can be characterized using the procedure below:

Definition 21 (Reduct $P/\mathcal{I}$)**.** Let $\mathcal{I}$ be an interpretation of an *NLP* P. The reduct of P with respect to $\mathcal{I}$, denoted $P/\mathcal{I}$, is the *NLP* constructed as follows:

a) remove any rule $r \in P$ such that $\mathit{body}^-(r) \cap \mathbf{t}(\mathcal{I}) \neq \emptyset$;

b) then, remove any occurrence of $\texttt{not}\ b_j$ from P such that $\mathcal{I}(b_j) = \mathbf{f}$;

c) lastly, replace any remaining occurrence of `not` b_j by a special atom **u** (**u** $\notin$ *HB*$_P$).

In the above procedure, **u** is assumed to be an atom not in *HB*$_P$ which is undefined in every interpretation of P. Note that $P/\mathcal{I}$ is a positive program, as all negative literals have been removed. As a consequence, $P/\mathcal{I}$ has a unique least 3-valued model (Przymusinski, 1990), obtained by the Ψ operator defined next:

Definition 22 (Ψ operator, Przymusinski 1990)**.** Let P be a positive program and $\mathcal{I}$ be an interpretation. Define $\Psi_P(\mathcal{I})$ as follows:

a) $c \in \mathbf{t}(\Psi_P(\mathcal{I}))$ iff $c \in$ *HB*$_P$ and there exists $c \leftarrow a_1, \ldots, a_m \in P$ such that for all i, $1 \leq i \leq m$, $a_i \in \mathbf{t}(\mathcal{I})$;

b) $c \in \mathbf{f}(\Psi_P(\mathcal{I}))$ iff $c \in$ *HB*$_P$ and for every $c \leftarrow a_1, \ldots, a_m \in P$, there exists i, $1 \leq i \leq m$, such that $a_i \in \mathbf{f}(\mathcal{I})$.

The least 3-valued model of $P/\mathcal{I}$ is given by $\Psi^{\uparrow\omega}_{P/\mathcal{I}}$ (Przymusinski, 1990), the least fixpoint of $\Psi_{P/\mathcal{I}}$ iteratively obtained as follows:

$$\begin{aligned}
\Psi^{\uparrow 0}_{P/\mathcal{I}} &= (\emptyset, HB_{P/\mathcal{I}}, \emptyset),\\
\Psi^{\uparrow i+1}_{P/\mathcal{I}} &= \Psi_{P/\mathcal{I}}(\Psi^{\uparrow i}_{P/\mathcal{I}}),\\
\mathbf{t}(\Psi^{\uparrow \omega}_{P/\mathcal{I}}) &= \bigcup_{i<\omega} \mathbf{t}(\Psi^{\uparrow i}_{P/\mathcal{I}}),\\
\mathbf{f}(\Psi^{\uparrow \omega}_{P/\mathcal{I}}) &= \bigcap_{i<\omega} \mathbf{f}(\Psi^{\uparrow i}_{P/\mathcal{I}}),
\end{aligned}$$

where ω denotes the first infinite ordinal.

The *NLP* semantics studied in this paper are introduced next:

Definition 23. Let $\mathcal{I}$ be an interpretation of an *NLP* P:

a) $\mathcal{I}$ is a partial stable model of P iff $\Psi^{\uparrow\omega}_{P/\mathcal{I}} = \mathcal{I}$ (Przymusinski, 1990);

b) $\mathcal{I}$ is a well-founded model of P iff $\mathcal{I}$ is a partial stable model of P and there is no partial stable model $\mathcal{I}'$ of P such that $\mathbf{t}(\mathcal{I}') \subset \mathbf{t}(\mathcal{I})$ (Przymusinski, 1990);

c) $\mathcal{I}$ is a regular model of P iff $\mathcal{I}$ is a partial stable model of P and there is no partial stable model $\mathcal{I}'$ of P such that $\mathbf{t}(\mathcal{I}) \subset \mathbf{t}(\mathcal{I}')$ (Eiter; Leone; Sacca, 1997);

d) $\mathcal{I}$ is a stable model of P iff $\mathcal{I}$ is a partial stable model of P and $\mathbf{u}(\mathcal{I}) = \emptyset$ (Przymusinski, 1990);

e) $\mathcal{I}$ is an L-stable model of P iff $\mathcal{I}$ is a partial stable model of P and there is no partial stable model $\mathcal{I}'$ of P such that $\mathbf{u}(\mathcal{I}') \subset \mathbf{u}(\mathcal{I})$ (Eiter; Leone; Sacca, 1997).

Some of these definitions are not standard in logic programming, but their equivalence to the original definitions is proved by Caminada; Sá, *et al.* (2015). This format helps relate *NLP* semantics to argumentation semantics.

Example 5. Consider the *NLP* P

$$
\begin{array}{ll}
a \leftarrow \texttt{not}\ b & b \leftarrow \texttt{not}\ a \\
c \leftarrow \texttt{not}\ a, \texttt{not}\ c & c \leftarrow \texttt{not}\ c, \texttt{not}\ d \\
d \leftarrow \texttt{not}\ d & e \leftarrow \texttt{not}\ b, \texttt{not}\ e.
\end{array}
$$

It has the following logic programming semantics:

a) partial stable models $\mathcal{M}_1 = (\emptyset, \emptyset, \{a, b, c, d, e\})$, $\mathcal{M}_2 = (\{a\}, \{b\}, \{c, d, e\})$, and $\mathcal{M}_3 = (\{b\}, \{a, e\}, \{c, d\})$;
b) well-founded model $\mathcal{M}_1$;
c) regular models $\mathcal{M}_2$ and $\mathcal{M}_3$;
d) no stable models;
e) L-stable model $\mathcal{M}_3$.

In the next chapter, we discuss existing *BAF* semantics and present a new one for *BAF*s, called β-semantics, based on a duality between the fundamental notions of defeat and defense. In posterior chapters, we show the β-semantics is deeply connected to the *SETAF* and *NLP* semantics considered here.

3 BIPOLAR ARGUMENTATION FRAMEWORKS WITH A DUAL RELATION BETWEEN DEFEAT AND DEFENSE

In this chapter, we propose a new semantics for BAFs, called β-semantics, and position some of its contributions to the landscape of existing interpretations of support. We show that our definitions collapse into the corresponding concepts proposed for AAFs when the support relation is ignored. Moreover, we prove the semantics β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable defined here for BAFs are generalizations of the corresponding semantics for AAFs. In the following chapters, we leverage the properties of the β-semantics to establish correspondences to the well-known semantics of NLPs and SETAFs.

Parts of this chapter appeared in the work of Alcântara; Cordeiro (2024).

3.1 Introduction

Different criteria for determining how conflicting sources of support and attack should be evaluated and combined has led to the development of a plethora of *BAF* (recall Definition 15) semantics. Additionally, some works (Potyka, 2021) have noticed an asymmetry in how existing approaches address attacks and supports. We believe that despite the numerous semantics, there are still perspectives in *BAF*s worthy of further exploration. Before introducing the β-semantics, we will review how existing *BAF* semantics diverge in their treatment of support.

One crucial aspect to consider is the role played by the support relation. There are at least three manners of interpreting this relation:

a) *deductive support*, in which an argument A supports an argument B if the acceptance of A implies the acceptance of B (Boella *et al.*, 2010);
b) *necessary support*, in which an argument A supports an argument B if the acceptance of B implies the acceptance of A (Nouioua; Risch, 2010; Nouioua; Risch, 2011);
c) *evidential support*, in which to be accepted, an argument must have support from a prima-facie argument, which is a special type of argument that can stand on its own without requiring support from other arguments (Oren; Norman, 2008;

Oren; Reed; Luck, 2010).

We illustrate next that the interpretation of support is controversial even for simple *BAF*s.

Example 6. Consider the two *BAF*s depicted in Figure 5. One could argue which arguments should be accepted in these frameworks.

(a) $\mathcal{B}_1$ (b) $\mathcal{B}_2$

Figure 5 – *BAF*s from Example 6.

In $\mathcal{B}_1$, with a deductive interpretation of the support relation as in the works of Boella *et al.* (2010); Cayrol; Lagasquie-Schiex (2013), we accept A_1 and A_3, and the acceptance of A_1 leads to the rejection of A_2. In contrast, with a necessary support according to Nouioua; Risch (2010, 2011), we accept only A_1, as the acceptance of A_3 would lead to the acceptance of the rejected argument A_2. Assuming the support relation is evidential (Oren; Norman, 2008; Oren; Reed; Luck, 2010) and that A_2 is the only *prima-facie* argument in $\mathcal{B}_1$, we can ignore the attack from A_1 to A_2 as A_1 is neither a *prima-facie* argument nor is supported by any argument. Thus A_1 is rejected and both A_2 and A_3 are accepted.

Within the same category, there may still be divergences. In $\mathcal{B}_2$, consider the deductive support as in the works of Boella *et al.* (2010); Cayrol; Lagasquie-Schiex (2013). In this case, only A_1 is accepted; the rejection of A_2 due to A_1's attack results in the rejection of A_3, as it supports A_2. However, if we consider the deductive support as in the deductive labelings introduced by Potyka (2020a), it is also possible (among other possibilities) to accept both A_2 and A_3 in $\mathcal{B}_2$. The rationale is that the acceptance of A_3 leads, by deduction, to accept A_2, which, in turn, results in the rejection of A_1. In yet another perspective, one could argue all arguments should be accepted: for A_1 and A_3, the reason is that they are not attacked at all; for A_2, it follows from A_3's support (which defends A_2 against the attack from A_1).

The last interpretation above is the one we propose in this chapter. It is clear from the previous example how the combination of support and attack is controversial, even in simple frameworks. In more complex settings, various possibilities of combining support and attack can induce the construction of complex forms of indirect attacks and supports. Consequently, inspired by variegated points of view, researchers have proposed many semantics for *BAF*s

(Boella *et al.*, 2010; Cayrol; Lagasquie-Schiex, 2007, 2013; Nouioua; Risch, 2010, 2011; Oren; Norman, 2008; Potyka, 2020a,b, 2021).

Several of these works have associated attacks with conflicts: an argument A conflicts with B if A attacks B or B attacks A. Indeed, this is the role played by conflicts in *AAF*s (Dung, 1995). As the existence of an attack from A to B is sufficient to indicate a conflict between A and B, if A attacks B, the acceptance of A implies the rejection of B. In this work, we propose new semantics for *BAF*s based on a generalization of the notion of conflict to involve both attacks and supports, where attacks contribute to the conflict and supports contribute to avoiding the conflict.

With this motivation, we interpret supports and attacks as described below:

a) we consider a deductive interpretation of the support relation, i.e., accepting A is enough to accept any argument supported by A;
b) an attack from an argument A to an argument B means that A brings evidence against B, but accepting A may not be enough to reject B, because B may be supported by other pieces of evidence;
c) an argument is accepted in the absence of evidence against it (this is also the case in *AAF*s).

As attacks are not enough, we need a stronger notion to associate with conflicts: defeat, which is a relation between a set of arguments and an argument and is intended to represent a successful collective attack. Similarly, we define defense as a successful collective support. Our aim is to provide a dual characterization of defeat and defense in the following sense: a set $\mathcal{S}$ of arguments defeats an argument A iff $\mathcal{S}$ defeats every supporter of A, and $\mathcal{S}$ defends A iff $\mathcal{S}$ defends every argument supported by A. The idea is that from the defeat of A, we obtain the defeat of every argument supporting A, and from the defense of A, we obtain the defense of every argument supported by A.

As our interpretation of support stems from the fundamentals notions of defeat and defense (w.r.t. some extension), for this chapter we will define the new semantics in both a extension-based and labeling-based characterization. Later, we show they are equivalent to each other. This helps clarify how we adapt Dung's (1995) essential concepts in *AAF*s to the context of support in argumentation. Let us illustrate our proposal with Examples 7 and 8.

Example 7 (Adapted from Cohen *et al.* 2014)**.** Consider a scenario where two soccer teams, Liverpool and Manchester United, are in the final race to win the Premier League. Suppose that Liverpool wins the Premiere League ($\mathcal{LPL}$) if it wins its last match ($\mathcal{LW}$) or Manchester United does not win its own ($\mathcal{MUNW}$). Note that Liverpool and Manchester United are not playing against each other and thus the results of their matches are independent. Suppose now that Liverpool loses its last match ($\mathcal{LL}$) and Manchester United does not win its own. Then, according to our proposal, we have that both $\mathcal{LW}$ and $\mathcal{MUNW}$ support $\mathcal{LPL}$, as accepting $\mathcal{LW}$ (or $\mathcal{MUNW}$) implies accepting $\mathcal{LPL}$. Furthermore, $\mathcal{LL}$ attacks both $\mathcal{LW}$ and $\mathcal{LPL}$ as $\mathcal{LL}$ brings evidence against them (see *BAF* $\mathcal{B}_3$ in Figure 6). $\mathcal{LL}$ is enough to defeat $\mathcal{LW}$, as accepting $\mathcal{LL}$ implies rejecting $\mathcal{LW}$. However, $\mathcal{LL}$ does not suffice to defeat $\mathcal{LPL}$, because $\mathcal{LPL}$ is supported by another piece of evidence ($\mathcal{MUNW}$). In this case, although $\mathcal{LL}$ attacks $\mathcal{LPL}$, there is no conflict between them, and it is reasonable to accept $\mathcal{LL}$, $\mathcal{LPL}$, and $\mathcal{MUNW}$.

(a) $\mathcal{B}_3$ (b) $\mathcal{B}_4$

Figure 6 – *BAF*s from Examples 7 and 8.

Now we explore the well-known Tweety example, where the support relation enhances the role played by the exception to a general rule.

Example 8 (Tweety)**.** Consider the following arguments:

a) A: Tweety cannot fly;
b) B: Tweety is a bird (and in general, birds can fly);
c) C: Tweety is a penguin.

According to our proposal, there is a mutual attack between A and B as each argument brings evidence against the other. Additionally, accepting C implies accepting A and B; so C supports both of them (see *BAF* $\mathcal{B}_4$ in Figure 6). Despite the mutual attack, neither A defeats B nor does B defeat A, because of the support provided by C. Note that C represents an exception to the general rule that birds usually fly (and if something cannot fly, then it usually is not a bird). Thus we can accept A, B, and C.

Taking profit from this dual perspective, we generalize to *BAF*s basic notions defined for *AAF*s as conflict-freeness, acceptable arguments, and admissible sets of arguments. These new definitions lead naturally to the characterization of new semantics corresponding to generalizations of the respective extensions for *AAF*s, namely the complete, preferred, grounded, stable, and semi-stable extensions. As a consequence, we not only generalize them but also generalize significant results proved by Dung (1995) to *BAF*s, including the Fundamental Lemma for *AAF*s.

In a similar vein, we show that our semantics for *BAF*s can be equivalently expressed as argument labelings while also generalizing results proven by Caminada (2006a); Caminada; Gabbay (2009). By generalizing these results and semantics in the extension-based and labeling-based settings, our proposal emerges as a natural candidate to provide semantics for *BAF*s.

This chapter is organized as follows: in Section 3.2, we present the extension-based semantics for *BAF*s and its fundamental results. Next, we introduce the labeling-based semantics for *BAF*s and show they are equivalent to the corresponding extension-based semantics defined in the previous section. In Section 3.4, we discuss additional motivations and intuitions behind the proposed semantics. In the sequel, we focus on related works and compare them with our approach. Finally, in Section 3.6, we conclude the chapter by summarizing the key findings and how they relate to the following chapters.

3.2 Extension-based β-semantics

We will follow Dung's (1995) approach, by first defining the fundamental notions of conflict-freeness, acceptable arguments, and admissible sets of arguments for *BAF*s. Then, we will define the corresponding extension-based semantics based on these notions.

Given a *BAF* (Definition 15), we are interested not only in the direct supporters of an argument but also in those indirect ones. We call them indistinctly the supporters of an argument:

Definition 24 (Supporters)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $A \in \mathcal{A}$ an argument. We define the supporters of A recursively as follows:

a) A is a supporter of A in $\mathcal{B}$;

b) if A' is a supporter of A in $\mathcal{B}$ and $(B, A') \in \mathit{Sup}$, then B is a supporter of A in $\mathcal{B}$.

By $\mathfrak{Sup}(A) = \{A' \in \mathcal{A} \mid A' \textit{ is a supporter of } A \textit{ in } \mathcal{B}\}$, we denote the set of all supporters of A in $\mathcal{B}$. We denote by $\mathfrak{Sup}$ the relation $\{(B, A) \mid B \in \mathfrak{Sup}(A)\}$. Alternatively, we can simply define $\mathfrak{Sup}$ as the reflexive and transitive closure of *Sup*.

We proceed by generalizing the notions of defeat and defense:

Definition 25 (Defeat and defense)**.** Let $\mathcal{S} \subseteq \mathcal{A}$ be a set of arguments in *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. We say the following:

a) $\mathcal{S}$ defeats A in $\mathcal{B}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in \mathit{Att}(A')$ such that $B \in \mathcal{S}$;

b) $\mathcal{S}$ defends A in $\mathcal{B}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in \mathit{Att}(A')$, it holds $\mathcal{S}$ defeats B in $\mathcal{B}$.

When the context is clear, we will omit the reference to $\mathcal{B}$ and state simply that $\mathcal{S}$ defeats/defends A.

We can comprehend defeat as a generalization of the notion of attack: for a set of arguments $\mathcal{S}$ to defeat an argument A, it does not suffice that an argument in $\mathcal{S}$ attacks A, but rather, every supporter of A has to be attacked by an argument in $\mathcal{S}$, leaving no possibility for A to be favored. In its turn, $\mathcal{S}$ defends A if there is a supporter A' of A for which every argument attacking A' is defeated by $\mathcal{S}$. Attentive readers will notice that this notion of defense aligns with the corresponding concept proposed by Dung (1995) for *AAF*s. The following result shows a dual behaviour between defeat and defense:

Proposition 3. Let $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$ in *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. The following holds:

a) $\mathcal{S}$ defeats A iff $\mathcal{S}$ defeats A' for every $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{S}$ defeats A'' for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$;

b) $\mathcal{S}$ defends A iff $\mathcal{S}$ defends A' for some $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Intuitively, from the defeat of A, we obtain the defeat of every argument supporting A, and from the defense of A, we obtain the defense of every argument supported by A. The next step is to associate conflicts with defeats rather than with attacks as in *AAF*s:

Definition 26 (Conflict-free sets)**.** Let $\mathcal{S} \subseteq \mathcal{A}$ be a set of arguments in *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. We say $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff for every argument $A \in \mathcal{S}$, it holds $\mathcal{S}$ does not defeat A.

In an *AAF*, the existence of an argument in $\mathcal{S}$ attacking an argument in $\mathcal{S}$ is enough for $\mathcal{S}$ to be non-conflict-free; on the other hand, in a *BAF*, $\mathcal{S}$ has to defeat an argument in $\mathcal{S}$ to be non-conflict-free. In the *BAF* below

$$A_1 \longrightarrow A_2 \dashrightarrow A_3 \longleftarrow A_4$$

the set $\mathcal{S}_1 = \{A_3, A_4\}$ is conflict-free, because although A_4 attacks A_3, $\mathcal{S}_1$ does not defeat A_3, as A_2 (a supporter of A_3) is not defeated by $\mathcal{S}_1$. The intuition here is that it is still possible to support A_3 via A_2 without conflicting with $\mathcal{S}_1$. In contradistinction, $\mathcal{S}_2 = \{A_1, A_3, A_4\}$ is non-conflict-free as $\mathcal{S}_2$ defeats A_3.

The definition of β-admissible arguments for *BAF*s is given as follows:

Definition 27 (β-admissible arguments)**.** Let $\mathcal{S} \subseteq \mathcal{A}$ be a set of arguments in *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. The characteristic function $F_{\mathcal{B}} : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ is defined as

$$F_{\mathcal{B}}(\mathcal{S}) = \{A \mid A \textit{ is defended by } \mathcal{S} \textit{ in } \mathcal{B}\}.$$

We can alternatively say A is *acceptable* w.r.t. $\mathcal{S}$ to mean that A is defended by $\mathcal{S}$, i.e., $A \in F_{\mathcal{B}}(\mathcal{S})$. A conflict-free set $\mathcal{S}$ in $\mathcal{B}$ is said to be β-admissible iff $\mathcal{S} \subseteq F_{\mathcal{B}}(\mathcal{S})$, which means that the arguments in the set can defend themselves against any attackers in the framework. Finally, we write

$$\mathcal{S}^+_{\mathcal{B}} = \{A \mid A \textit{ is defeated by } \mathcal{S} \textit{ in } \mathcal{B}\}$$

to refer to the set of arguments defeated by $\mathcal{S}$ in $\mathcal{B}$.

With these definitions in mind, we can present a generalization of those semantics for *AAF*s to *BAF*s as follows:

Definition 28 (Extension-based β-semantics)**.** Given a *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{S} \subseteq \mathcal{A}$, the following holds:

a) $\mathcal{S}$ is β-complete iff $\mathcal{S}$ is a conflict-free fixpoint of $F_{\mathcal{B}}$, i.e., $\mathcal{S}$ is conflict-free and $\mathcal{S} = F_{\mathcal{B}}(\mathcal{S})$;
b) $\mathcal{S}$ is β-grounded iff $\mathcal{S}$ is a $\subseteq$-minimal β-complete extension of $\mathcal{B}$;
c) $\mathcal{S}$ is β-preferred iff $\mathcal{S}$ is a $\subseteq$-maximal β-complete extension of $\mathcal{B}$;
d) $\mathcal{S}$ is β-stable iff $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ such that $\mathcal{S} \cup \mathcal{S}^+ = \mathcal{A}$;
e) $\mathcal{S}$ is β-semi-stable iff $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ with $\subseteq$-maximal $\mathcal{S} \cup \mathcal{S}^+$.

From Definition 28, it directly follows that every β-grounded, β-preferred, β-stable, and β-semi-stable extension of $\mathcal{B}$ is also a β-complete extension of $\mathcal{B}$. Indeed, we emphasize that these extension-based semantics are defined analogously to the corresponding extension-based semantics for *AAF*s (see Definition 12). Let us illustrate these semantics with an example:

Example 9. Consider the *BAF* in Figure 7. Its semantics are presented below:

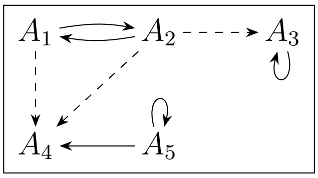


Figure 7 – *BAF* from Example 9.

a) β-complete extensions $\emptyset$, $\{A_1, A_4\}$ and $\{A_2, A_3, A_4\}$;

b) β-grounded extension $\emptyset$;

c) β-preferred extensions $\{A_1, A_4\}$ and $\{A_2, A_3, A_4\}$;

d) no β-stable extensions;

e) β-semi-stable extension $\{A_2, A_3, A_4\}$.

The next two results (Theorem 4 and Corollary 5) guarantee some equivalence results between *BAF*s $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}^* = (\mathcal{A}, \mathit{Att}, \mathit{Sup}^*)$, where Sup^* is the reflexive and transitive closure of *Sup*:

Theorem 4. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}^* = (\mathcal{A}, \mathit{Att}, \mathit{Sup}^*)$ be *BAF*s such that Sup^* is the reflexive and transitive closure of *Sup*, i.e., $\mathit{Sup}^* = \mathfrak{Sup}$. For any set $\mathcal{S} \subseteq \mathcal{A}$, the following holds:

a) $F_{\mathcal{B}}(\mathcal{S}) = F_{\mathcal{B}^*}(\mathcal{S})$;

b) $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff $\mathcal{S}$ is conflict-free in $\mathcal{B}^*$.

Corollary 5. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}^* = (\mathcal{A}, \mathit{Att}, \mathit{Sup}^*)$ be *BAF*s such that $\mathit{Sup}^* = \mathfrak{Sup}$. The following holds:

a) $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}^*$;

b) $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-complete extension of $\mathcal{B}^*$;

c) $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}^*$;

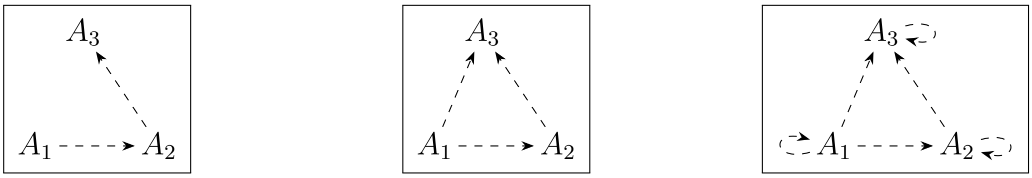
d) $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}^*$;

e) $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a a β-stable extension of $\mathcal{B}^*$;

f) $\mathcal{S}$ is a β-semi-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a a β-semi-stable extension of $\mathcal{B}^*$.

This means we can use the support relation *Sup* or its reflexive and transitive closure interchangeably in our approach. The *BAF*s $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ depicted in Figure 8 are considered semantically equivalent.

Now we concentrate on proving that our proposal is a natural generalization of the concepts defined by Dung (1995) for *AAF*s. In the sequel, we demonstrate that the notions of



(a) $\mathcal{B}_1$ (b) $\mathcal{B}_2$ (c) $\mathcal{B}_3$

Figure 8 – Semantically equivalent *BAF*s.

defeat, defense, and conflict-freeness introduced here for *BAF*s are respectively generalizations of defeat, defense, and conflict-freeness in *AAF*s:

Proposition 6. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* such that $Sup = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, Att)$ be the associated *AAF*. For $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$, the following holds:

a) $\mathcal{S}$ defeats A in $\mathcal{B}$ iff $\mathcal{S}$ defeats A in $\mathfrak{F}$;

b) $\mathcal{S}$ defends A in $\mathcal{B}$ iff $\mathcal{S}$ defends A in $\mathfrak{F}$;

c) $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff $\mathcal{S}$ is conflict-free in $\mathfrak{F}$.

From Proposition 6, we obtain straightforwardly that β-admissible, β-complete, β-grounded, β-stable and β-semi-stable extensions for *BAF*s are generalizations, respectively, of admissible, complete, grounded, stable, and semi-stable extensions defined for *AAF*s:

Corollary 7. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* such that $Sup = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, Att)$ be the corresponding *AAF*. For $\mathcal{S} \subseteq \mathcal{A}$, the following holds:

a) $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}$ iff $\mathcal{S}$ is an admissible extension of $\mathfrak{F}$;

b) $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ iff $\mathcal{S}$ is a complete extension of $\mathfrak{F}$;

c) $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}$ iff $\mathcal{S}$ is a grounded extension of $\mathfrak{F}$;

d) $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a preferred extension of $\mathfrak{F}$;

e) $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a stable extension of $\mathfrak{F}$;

f) $\mathcal{S}$ is a β-semi-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a semi-stable extension of $\mathfrak{F}$.

Besides generalizing these *AAF* semantics to *BAF*s, we will show in the remainder of this section that our proposal also preserves crucial results proved by Dung (1995) for *AAF*s. With this purpose, we follow the same path, starting with the generalization of Dung's Fundamental Lemma to *BAF*s:

Lemma 8 (Fundamental Lemma)**.** Let $\mathcal{B}$ be a *BAF*, $\mathcal{S}$ be a β-admissible set of arguments, and A, A' be arguments which are acceptable with respect to $\mathcal{S}$. Then, the following holds:

a) $\mathcal{S}' = \mathcal{S} \cup \{A\}$ is β-admissible;

b) A' is acceptable with respect to $\mathcal{S}'$.

Just as in the case of *AAF*s (Dung, 1995), from the Fundamental Lemma it follows a strong characterization of β-admissible extensions:

Theorem 9. Let $\mathcal{B}$ be a *BAF*. The following holds:

a) the set of all β-admissible sets of $\mathcal{B}$ forms a complete partial order with respect to set inclusion;
b) for each β-admissible extension $\mathcal{S}$ of $\mathcal{B}$, there exists a maximal β-admissible extension $\mathcal{E}$ of $\mathcal{B}$ such that $\mathcal{S} \subseteq \mathcal{E}$.

Theorem 9 applied to the fact that the empty set is always β-admissible implies the existence of some $\subseteq$-maximal β-admissible extension.

Corollary 10. Every *BAF* possesses at least one $\subseteq$-maximal β-admissible extension.

Dung (1995) shows that a set of arguments $\mathcal{S}$ is a stable extension if $\mathcal{S}$ is the set of arguments (using our terminology) not defeated by $\mathcal{S}$. Furthermore, in this same work, he shows that every stable extension is a preferred extension, but not vice versa. In the sequel, we show these properties are preserved in *BAF*s with β-stable and β-preferred extensions:

Proposition 11. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. Then $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S} = \{A \mid A \textit{ is not defeated by } \mathcal{S}\}$.

Proposition 12. For any *BAF* $\mathcal{B}$, any β-stable extension of $\mathcal{B}$ is a β-preferred extension of $\mathcal{B}$. However, it is not always the case that every β-preferred extension of $\mathcal{B}$ is a β-stable extension of $\mathcal{B}$.

Lemmas 13 and 14 are related to the $F_{\mathcal{B}}$ operator: Lemma 13 guarantees $F_{\mathcal{B}}$ preserves the conflict-freeness property, while Lemma 14 shows $F_{\mathcal{B}}$ is monotonic.

Lemma 13. Let $\mathcal{B}$ be a *BAF*. If $\mathcal{S}$ is a conflict-free set in $\mathcal{B}$, then $F_{\mathcal{B}}(\mathcal{S})$ is also conflict-free in $\mathcal{B}$.

Lemma 14. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. Then $F_{\mathcal{B}}(\mathcal{S})$ is monotonic with respect to set inclusion.

Now we can guarantee an alternative characterization of β-preferred extensions (Lemma 15) as well as the existence of β-preferred/β-complete extensions (Theorem 16):

Lemma 15. Let $\mathcal{B}$ be a *BAF*. It holds $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a $\subseteq$-maximal β-admissible extension of $\mathcal{B}$.

The next result is an immediate consequence of Corollary 10, Lemma 15, and Definition 28:

Theorem 16. Every *BAF* has at least one β-preferred/β-complete extension.

We also have that the β-grounded extension is uniquely defined for every *BAF* $\mathcal{B}$ and it coincides with the least fixpoint of $F_{\mathcal{B}}$:

Theorem 17. Every *BAF* $\mathcal{B}$ possesses a unique β-grounded extension and it is the $\subseteq$-least fixpoint of $F_{\mathcal{B}}$.

Just like complete extensions (Dung, 1995), β-complete extensions also constitute a complete semilattice under set inclusion:

Theorem 18. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. The β-complete extensions of $\mathcal{B}$ form a complete semilattice with respect to set inclusion.

This sequence of results show that our proposal preserves many of the concepts and properties in the foundantion of *AAF*s. In the next section, we proceed with this comparison to *AAF*s by focusing on labeling-based semantics.

3.3 Labeling-based β-semantics

We present in the current section an alternative characterization of the semantics for *BAF*s as labelings. Our strategy involves generalizing the definitions of labeling-based semantics for *AAF*s (see Caminada, 2006; Caminada; Gabbay, 2009) to the context of *BAF*s while also preserving results proved there. Then, we show the equivalence between the extension-based and labeling-based semantics proposed in our work.

Just as in *AAF*s, a labeling is a total function $\mathcal{L} : \mathcal{A} \rightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$. Inspired by Caminada (2006a); Caminada; Gabbay (2009), we define β-admissible, β-complete, β-grounded, β-preferred, β-stable and β-semi-stable labelings as follows:

Definition 29 (β-admissible labelings)**.** A labeling $\mathcal{L}$ is a β-admissible labeling of *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$ if for any $A \in \mathcal{A}$, the following holds:

a) if $\mathcal{L}(A) = \mathtt{in}$, then there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $\mathcal{L}(B) = \mathtt{out}$;

b) if $\mathcal{L}(A) = \mathtt{out}$, then for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in Att(A')$ such that $\mathcal{L}(B) = \mathtt{in}$.

Definition 30 (Labeling-based β-semantics)**.** A labeling $\mathcal{L}$ is a β-complete labeling of *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$ if for any $A \in \mathcal{A}$, the following holds:

a) $\mathcal{L}(A) = \mathtt{in}$ if and only if there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $\mathcal{L}(B) = \mathtt{out}$;

b) $\mathcal{L}(A) = \mathtt{out}$ if and only if for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in Att(A')$ such that $\mathcal{L}(B) = \mathtt{in}$.

Additionally, a β-complete labeling $\mathcal{L}$ of $\mathcal{B}$ is said to be the following:

a) β-grounded, if $\mathtt{in}(\mathcal{L})$ is $\subseteq$-minimal among the β-complete labelings of $\mathcal{B}$;

b) β-preferred, if $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B}$;

c) β-stable, if $\mathtt{undec}(\mathcal{L}) = \emptyset$;

d) β-semi-stable, if $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among the β-complete labelings of $\mathcal{B}$.

When convenient, we reuse the notation of writing a labeling $\mathcal{L}$ as the triple $(\mathtt{in}(\mathcal{L}), \mathtt{out}(\mathcal{L}), \mathtt{undec}(\mathcal{L}))$. Now we recall the *BAF* exhibited in Figure 7 and display it again in Figure 9 for easier reference.

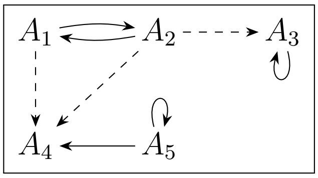


Figure 9 – *BAF* exhibited in Figure 7.

Example 10. Consider the *BAF* in Figure 9. It has the following labeling-based semantics:

a) β-complete labelings $\mathcal{L}_1 = (\emptyset, \emptyset, \mathcal{A})$, $\mathcal{L}_2 = (\{A_1, A_4\}, \{A_2\}, \{A_3, A_5\})$, and $\mathcal{L}_3 = (\{A_2, A_3, A_4\}, \{A_1\}, \{A_5\})$;

b) β-grounded labeling $\mathcal{L}_1$;

c) β-preferred labelings $\mathcal{L}_2$ and $\mathcal{L}_3$;

d) no β-stable labelings;

e) β-semi-stable labeling $\mathcal{L}_3$.

Note the correspondence between the extension-based semantics obtained in Example 9 and their counterpart as labeling-based semantics obtained here. Further in this section, we will show the equivalence of these approaches. Next, in Propositions 19 and 20, we exploit properties of the β-complete labelings:

Proposition 19. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathcal{L}$ be a β-complete labeling of $\mathcal{B}$. For any $A \in \mathcal{A}$, the following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A'') = \texttt{in}$ for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$;

b) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A'') = \texttt{out}$ for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Proposition 20. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathcal{L}$ a labeling of $\mathcal{B}$. Then, $\mathcal{L}$ is β-complete iff for any $A \in \mathcal{A}$, all conditions below are satisfied:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\texttt{in}(\mathcal{L})$ defends A;

b) $\mathcal{L}(A) = \texttt{out}$ iff $\texttt{in}(\mathcal{L})$ defeats A.

When analyzed together, Propositions 19 and 20 evince the dual nature of defeat and defense and are a consequence of Proposition 3. Now we will show that the labeling-based semantics for *BAF* defined here are generalizations of the corresponding semantics for *AAF*s.

Proposition 21. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* such that $Sup = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, Att)$ the corresponding *AAF*. For a labeling $\mathcal{L}$ of $\mathcal{B}$, we can guarantee the following properties:

a) $\mathcal{L}$ is a β-admissible labeling of $\mathcal{B}$ iff $\mathcal{L}$ is an admissible labeling of $\mathfrak{F}$;

b) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{F}$;

c) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a grounded labeling of $\mathfrak{F}$;

d) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a preferred labeling of $\mathfrak{F}$;

e) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a stable labeling of $\mathfrak{F}$;

f) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{F}$.

Next, we show the equivalence between the extension-based semantics and labeling-based semantics, while in Subsection 3.3.2 we investigate results involving minimization and maximization of labelings.

3.3.1 Equivalence between extension-based and labeling-based β-semantics

We will shift our attention to show that there exists a one-to-one correspondence between the labeling-based and extension-based semantics for *BAF*s. With this intention, we define two functions: $\texttt{Ext2Lab}_{\mathcal{B}}$ and $\texttt{Lab2Ext}_{\mathcal{B}}$. The function $\texttt{Ext2Lab}_{\mathcal{B}}$ takes a conflict-free set of arguments as input and converts it to a labeling. Conversely, $\texttt{Lab2Ext}_{\mathcal{B}}$ takes a labeling as input and converts it to a set of arguments.

Definition 31. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*, $\mathcal{S} \subseteq \mathcal{A}$ a conflict-free set of arguments, and $\mathcal{L}$ a labeling of $\mathcal{B}$. We define the following:

a) $\texttt{Ext2Lab}_{\mathcal{B}}(\mathcal{S}) = \mathcal{L}'$ such that all conditions below hold:
   - $\mathcal{L}'(A) = \texttt{in}$ iff $A \in \mathcal{S}$;
   - $\mathcal{L}'(A) = \texttt{out}$ iff $A \in \mathcal{S}^+$;
   - $\mathcal{L}'(A) = \texttt{undec}$ iff $A \notin (\mathcal{S} \cup \mathcal{S}^+)$;

b) $\texttt{Lab2Ext}_{\mathcal{B}}(\mathcal{L}) = \texttt{in}(\mathcal{L})$.

In fact, the functions $\texttt{Ext2Lab}_{\mathcal{B}}$ and $\texttt{Lab2Ext}_{\mathcal{B}}$ have been derived by Caminada (2006a); Caminada; Gabbay (2009), but have been adapted here to accommodate *BAF*s. Occasionally, when the context is clear, we will use the notations $\texttt{Ext2Lab}$ and $\texttt{Lab2Ext}$ instead of $\texttt{Ext2Lab}_{\mathcal{B}}$ and $\texttt{Lab2Ext}_{\mathcal{B}}$, respectively.

The first result of this subsection exhibits two relations involving β-admissible labelings and β-admissible sets:

Theorem 22. Let $\mathcal{B}$ be a *BAF*. Then the following holds:

a) if $\mathcal{L}$ is a β-admissible labeling of $\mathcal{B}$, then $\texttt{Lab2Ext}(\mathcal{L})$ is a β-admissible set of $\mathcal{B}$;

b) if $\mathcal{E}$ is a β-admissible set of $\mathcal{B}$, then $\texttt{Ext2Lab}(\mathcal{E})$ is a β-admissible labeling of $\mathcal{B}$.

These functions are closely related, as shown next.

Theorem 23. When restricting their domains respectively to β-complete extensions and β-complete labelings, the functions $\texttt{Ext2Lab}$ and $\texttt{Lab2Ext}$ are bijections and each other's inverse.

We proceed by showing a one-to-one correspondence between labelings and extensions in the β-semantics.

Theorem 24. Let $\mathcal{B}$ be a *BAF*. Then the following holds:

a) $\mathcal{L}$ is a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable labeling of $\mathcal{B}$ iff $\texttt{Lab2Ext}(\mathcal{L})$ is respectively a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable extension of $\mathcal{B}$;
b) $\mathcal{E}$ is a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable extension of $\mathcal{B}$ iff $\texttt{Ext2Lab}(\mathcal{E})$ is respectively a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable labeling of $\mathcal{B}$.

This correspondence between β-complete labelings and β-complete extensions suffices to guarantee the existence of a β-complete labeling for any *BAF* (Theorem 25) and the unicity of the minimal β-complete labeling of a *BAF*, i.e., any *BAF* has a unique β-grounded labeling (Theorem 26).

Theorem 25. For any *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$, there always exists a β-complete labeling of $\mathcal{B}$.

Theorem 26. Let $\mathcal{B}$ be a *BAF*. The β-complete labeling $\mathcal{L}$ of $\mathcal{B}$ where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ is unique.

3.3.2 On the minimization and maximization of argument labelings

In this subsection, we show that the same results involving minimization and maximization of complete labelings for *AAF*s (Caminada; Sá, *et al.*, 2015) also hold for β-complete labelings of *BAF*s.

Lemma 27. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$;
b) $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2)$.

Lemma 28. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) If $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subseteq \texttt{undec}(\mathcal{L}_1)$;
b) If $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subset \texttt{undec}(\mathcal{L}_1)$;
c) if $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subseteq \texttt{undec}(\mathcal{L}_1)$;
d) if $\texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subset \texttt{undec}(\mathcal{L}_1)$.

Lemma 29. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) if $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$, then $\mathcal{L}_1 = \mathcal{L}_2$;

b) if $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$, then $\mathcal{L}_1 = \mathcal{L}_2$.

From Lemmas 27 and 28, we obtain the Theorems 30, 31 and 32:

Theorem 30. Let $\mathcal{B}$ be a *BAF*. The following statements are equivalent:

a) $\mathcal{L}$ is a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$;

b) $\mathcal{L}$ is a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$;

c) $\mathcal{L}$ is a β-complete labeling where $\texttt{undec}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

Theorem 31. Let $\mathcal{B}$ be a *BAF*. It holds $\mathcal{L}$ is a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

Theorem 32. Let $\mathcal{B}$ be a *BAF*. If $\mathcal{L}$ is a β-complete labeling where $\texttt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$, then $\texttt{in}(\mathcal{L})$ and $\texttt{out}(\mathcal{L})$ are $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

In general, it is not the case that a maximal $\texttt{in}(\mathcal{L})$ (or $\texttt{out}(\mathcal{L})$) among all β-complete labelings corresponds to a minimal $\texttt{undec}(\mathcal{L})$ as we can see in the following example adapted from the work of Caminada (2006b):

Example 11. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* with $\mathcal{A} = \{A, B, C, D, E, F\}$, $Att = \{(A, B), (B, A), (B, C), (C, D), (D, E), (E, C)\}$ and $Sup = \{(E, F)\}$. The β-complete labelings of $\mathcal{B}$ are shown below:

a) $\mathcal{L}_1 = \{\{F\}, \emptyset, \{A, B, C, D, E\}\}$;

b) $\mathcal{L}_2 = \{\{A, F\}, \{B\}, \{C, D, E\}\}$;

c) $\mathcal{L}_3 = \{\{B, D, F\}, \{A, C, E\}, \emptyset\}$.

Note $\texttt{in}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_2)$ are $\subseteq$-maximal, but $\texttt{undec}(\mathcal{L}_2)$ is not $\subseteq$-minimal, as $\texttt{undec}(\mathcal{L}_3) \subset \texttt{undec}(\mathcal{L}_2)$.

The β-complete labelings with maximal in (*β-preferred labelings*) coincide with the β-complete labelings with maximal out (Theorem 31). Furthermore, the unique complete labeling where in is minimal (Theorem 26), i.e., the *β-grounded labeling*, is the same as the unique β-complete labeling where out is minimal (Theorem 30) and the same as the unique β-complete labeling where undec is maximal (Theorem 30). It is clear that (i) every β-stable labeling is also β-semi-stable and (ii) every β-semi-stable labeling is also β-preferred (Theorem 32). These results are summarized in Table 1.

Condition	Resulting labeling
NONE	β-complete
MAX `in`	β-preferred
MAX `out`	β-preferred
MAX `undec`	β-grounded
MIN `in`	β-grounded
MIN `out`	β-grounded
MIN `undec`	β-semi-stable
NO `undec`	β-stable

Table 1 – Effect of minimization/maximization of labels.

Notice the relations outlined in Table 1 are the same as those verified in the labeling-based semantics for *AAF*s (Caminada, 2006; Caminada; Gabbay, 2009). So far, we have shown that our proposal preserves many of the results obtained for *AAF*s. Next, we focus on the motivations and intuitions behind the proposed semantics, and how it can be applied in practice.

3.4 Discussion

We discuss throughout this section the motivations and intuitions underpinning the proposed semantics in the context of *BAF*s. After showing that we interpret the support relation as deductive, we examine whether the notion of defeat introduced in Definition 25 is not very strong. Then we offer insights into the ideas behind our support relation. We use three real-world examples to model scenarios in which our approach is suitable. Each scenario involves a distinct combination of attacks and supports, establishing a possible meaning to the role played by the support relation. We begin by guaranteeing the deductive character of our support relation:

Proposition 33. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathcal{S} \subseteq \mathcal{A}$ a set of arguments. If $A \in \mathcal{A}$ is acceptable with respect to $\mathcal{S}$, then B is acceptable with respect to $\mathcal{S}$ for every $B \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(B)$.

This result is immediate from Proposition 3, as A being acceptable w.r.t. $\mathcal{S}$ is simply another way of saying that $\mathcal{S}$ defends A. In particular, Proposition 3 states that $\mathcal{S}$ defends A iff $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

(a) $\mathcal{B}_1$ (b) $\mathcal{B}_2$

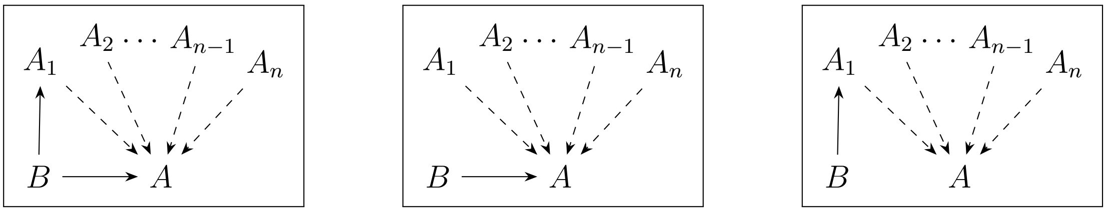
Figure 10 – *BAF*s from Example 12.

One could argue that the notion of defeat introduced in Definition 25 is too strong, as it requires every transitive supporter of an argument to be defeated in order to defeat an argument. In response, let us revisit Example 7 with an additional argument:

Example 12 (Continuing from Example 7)**.** Besides the arguments Liverpool wins the Premiere League ($\mathcal{LPL}$), Liverpool wins its last match ($\mathcal{LW}$), Manchester United does not win its last match ($\mathcal{MUNW}$), and Liverpool loses its last match ($\mathcal{LL}$), we include the argument Liverpool wins its last match or Manchester United does not win its last match ($\mathcal{LOM}$). As before, Liverpool wins the Premiere League if it wins its last match or Manchester United does not win its own. According to our understanding of the (deductive) support relation, both *BAF*s $\mathcal{B}_1$ and $\mathcal{B}_2$ in Figure 10 can equivalently model this scenario. In $\mathcal{B}_2$, the support relation from $\mathcal{B}_1$ is modeled with explicit transitive supports; to employ $\mathcal{B}_1$ or $\mathcal{B}_2$ is a question of design choice. In order to defeat $\mathcal{LPL}$ in $\mathcal{B}_1$, we have to defeat every transitive supporter of $\mathcal{LPL}$; in $\mathcal{B}_2$, however, we have to defeat only the direct supporters of $\mathcal{LPL}$ (besides $\mathcal{LPL}$ itself), because the transitive closure of the support relation has been applied in the design of the framework. Someone worried about the idea of defeating every transitive supporter of $\mathcal{LPL}$ to defeat $\mathcal{LPL}$ could prefer $\mathcal{B}_2$ as it considers only the direct supporters. However, in both *BAF*s, the arguments required to defeat $\mathcal{LPL}$ are the same.

Notice that this transitivity in the characterization of the defeat comes from the transitivity of the support relation: if argument A supports B, and B supports C, then A (indirectly) supports C, because by the deductive character of the support relation (Proposition 33), accepting A implies accepting B, which in turn implies accepting C. Although it is reasonable to look for weaker versions, we argue that our notion of defeat couples with the deductive (and transitive) nature of the support relation employed in our work. Besides, as we have already seen, this definition of defeat is crucial to guarantee important results as those related to generalizing properties of *AAF*s.

Another fundamental aspect to discuss is the intuitive meaning of the support relation. In the sequel, we will exploit the three scenarios depicted in Figure 11. We intend to motivate that depending on the kind of interation between support and attack, the support relation can play a distinct role.



(a) $\mathcal{B}_1$ (b) $\mathcal{B}_2$ (c) $\mathcal{B}_3$

Figure 11 – Three scenarios depicting the role of the support relation.

In the three *BAF*s of Figure 11, arguments $A_1, A_2, \ldots, A_{n-1}$, and A_n support A. In $\mathcal{B}_1$ (Figure 11a), argument B attacks both A and A_1; in $\mathcal{B}_2$ (Figure 11b), argument B attacks only A; in $\mathcal{B}_3$ (Figure 11c), argument B attacks only A_1. Let us now exploit these three scenarios with real-world examples and possible interpretations of the support relation:

a) one possibility for the first scenario (Figure 11a) is that the rejection of A_1 does not contribute to the acceptance of any $A_i \in \{A_2, \ldots, A_n\}$. When B attacks A_1, it attacks a pillar sustaining A and brings evidence against the acceptance of A; in other words, it also attacks A. Such an attack can be interpreted as a secondary attack on A, but without necessarily defeating it, as other arguments may support A. This is the scenario of Example 7, where the argument Liverpool wins the Premiere League ($\mathcal{LPL}$) is supported by two arguments: Liverpool wins the last match ($\mathcal{LW}$) and Manchester United does not win its own ($\mathcal{MUNW}$). Note the rejection of $\mathcal{LW}$ does not interfere in the acceptance of $\mathcal{MUNW}$. Then the argument Liverpool loses its last match ($\mathcal{LL}$) attacks both $\mathcal{LW}$ and $\mathcal{LPL}$.
b) in the second scenario (Figure 11b), we can interpret the support given by A_1, A_2, $\ldots$, A_{n-1}, A_n to A as an exception to the rule that B is normally not A. This is the case of Example 8, where argument C (Tweety is a penguin) represents an exception to the general rule that birds usually fly. The argument B (*Tweety is a bird*) attacks A (Tweety cannot fly), but it does not attack C.
c) in Figure 11c, we can consider that the acceptance of B contributes both positively and negatively with the acceptance of A. Hence, when B attacks A_1, we cannot say that B brings evidence against A, as it also has elements favoring

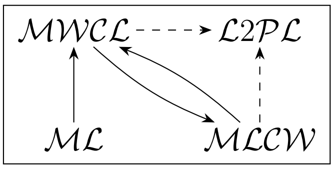


Figure 12 – *BAF* from the third scenario (Figure 11c).

A. We illustrate this scenario with the following example: suppose that Liverpool will finish in second place in the Premier League ($\mathcal{L}2\mathcal{PL}$) if $\mathcal{MWCL}$ or $\mathcal{MLCW}$, where $\mathcal{MWCL}$ is the argument Manchester United wins its last match and Chelsea does not win its last match, and $\mathcal{MLCW}$ is the argument Manchester United does not win its last match and Chelsea wins its last match. Assume that Manchester United loses its last match $\mathcal{ML}$. We have that $\mathcal{MWCL}$ and $\mathcal{MLCW}$ attack each other, and also support $\mathcal{L}2\mathcal{PL}$. In addition, $\mathcal{ML}$ attacks $\mathcal{MWCL}$. However, $\mathcal{ML}$ does not bring evidence against $\mathcal{L}2\mathcal{PL}$ as $\mathcal{ML}$ also contributes favorably with $\mathcal{MLCW}$, which is a supporter of $\mathcal{L}2\mathcal{PL}$. Because of this dubious behaviour, $\mathcal{ML}$ attacks $\mathcal{MWCL}$, but it does not attack $\mathcal{L}2\mathcal{PL}$. The resulting *BAF* is depicted in Figure 12.

Next, we contrast our interpretation of support to existing *BAF* semantics from the literature.

3.5 Related works

Several research efforts have yielded the development of different semantics for *BAF*s. One of the pioneering works in the area of *BAF*s is the paper by Cayrol; Lagasquie-Schiex (2005). They extend *AAF*s by taking into account two independent kinds of interaction between arguments: a defeat relation and a support relation. The associated semantics are based on a combination of the attack and support relations, resulting in new complex defeat relations. For a *BAF* $(\mathcal{A}, Att, Sup)$, they define two notions of defeat:

- a *supported defeat* from A to B is a direct attack $(A, B) \in Att$ or a sequence $(A_1, A_2, \ldots, A_{n-1}, A_n)$, $n \geq 3$, with $A_1 = A$, $A_n = B$, $(A_{n-1}, A_n) \in Att$ and $\forall i \in \{1, 2, \ldots, n-2\}$, it holds $(A_i, A_{i+1}) \in Sup$;
- a *secondary defeat*[1] from A to B is a sequence $(A_1, A_2, \ldots, A_{n-1}, A_n)$, $n \geq 3$, with $A_1 = A$, $A_n = B$, $(A_1, A_2) \in Att$ and $\forall i \in \{2, \ldots, n-1\}$, it holds $(A_i, A_{i+1}) \in Sup$.

[1] Cayrol; Lagasquie-Schiex (2005) employ the terminology *indirect defeat*, but in later works, they consider the terminology *secondary defeat*.

For $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$, they say $\mathcal{S}$ *set-defeats* A iff there exists a supported defeat or a secondary defeat for A from an element of $\mathcal{S}$. As can be seen in Figure 13, the inclusion of secondary defeats allows a necessary interpretation of the support relation: the rejection of argument A_2 is sufficient to reject the supported argument A_3. Therefore, any attacker of A_2 can also defeat A_3, and that is encoded by the secondary defeat (A_1, A_2, A_3). In our approach, support has a deductive interpretation. Hence, in Figure 13, it is possible to accept A_1 and A_3 at the same time.

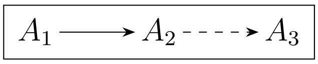


Figure 13 – Simple *BAF* illustrating secondary defeat from A_1 to A_3.

The notions of defense and conflict-freeness proposed by Dung (1995) are generalized to embody the concept of set-defeat; for a set $\mathcal{S} \subseteq \mathcal{A}$, they define defense and conflict-freeness as follows:

a) $\mathcal{S}$ CLS-*defends* A iff $\forall B \in \mathcal{A}$, if $\{B\}$ set-defeats A, then $\exists C \in \mathcal{S}$ such that $\{C\}$ set-defeats B;

b) $\mathcal{S}$ is CLS-*conflict-free* iff $\nexists A, B \in \mathcal{S}$ such that $\{A\}$ set-defeats B.

We introduced CLS (reference to Cayrol; Lagasquie-Schiex) in the definitions above to distinguish them from ours. A notable distinction is that their definition of conflict-freeness is stricter than the one proposed by Dung, while ours is more general:

a) if $\mathcal{S}$ is CLS-conflict-free, then $\mathcal{S}$ is conflict-free in Dung's sense;

b) if $\mathcal{S}$ is conflict-free in Dung's sense, then it is in our sense.

In the *BAF* depicted in Figure 14, the sets $\{A, G\}$ and $\{C, E\}$ are not CLS-conflict-free, but they are conflict-free both in Dung's sense and in our sense. On the other hand, the set $\{D, C\}$ is neither CLS-conflict-free nor conflict-free in Dung's sense, but it is in our sense.

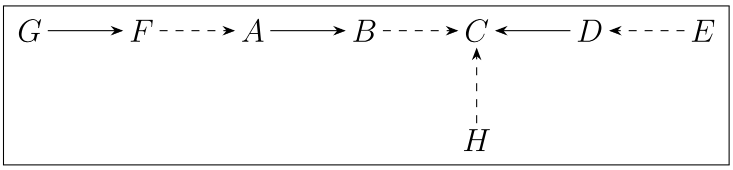


Figure 14 – *BAF* illustrating the semantics from Cayrol; Lagasquie-Schiex (2005).

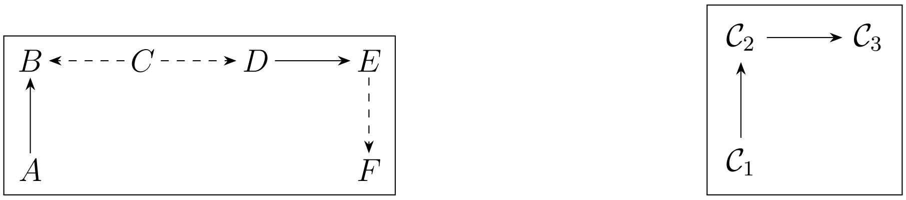
Recall Example 8, where there are three arguments: A (Tweety cannot fly); B (Tweety is a bird and in general, birds can fly); C (Tweety is a penguin). The expected answer is that A, B, and C are acceptable. This means that according to Cayrol; Lagasquie-Schiex's

(2005) semantics, the framework for encoding this scenario cannot contain attacks between these arguments. This encoding does not capture the nuance that A and B are in conflict (in general) and this conflict can be avoided by an exception (represented by C). This applies to all semantics we discuss in the current section that have a stricter notion of conflict-freeness than that of Dung's.

In their work, Cayrol; Lagasquie-Schiex (2005) propose three definitions for the notion of admissibility: d-admissible (more general), s-admissible, and c-admissible (most specific). In common, any extension under these semantics must be CLS-conflict-free and CLS-defend all its elements. Recalling the *BAF* in Figure 14, we observe that no set containing C is c-admissible, s-admissible, or d-admissible. The attack from D to C does not allow accepting both C and D. In addition, C cannot be accepted due to its lack of CLS-defense against the attack by D. In these semantics, the attack relation is stronger than the support relation, resulting in the rejection of C even though it is supported by the argument H, which is not attacked. Compared with our proposal, the support provided by the unattacked argument H is sufficient for accepting C even when accepting D. In general terms, a remarkable difference is that to a set $\mathcal{S}$ set-defeat an argument A, it is enough that an element of $\mathcal{S}$ attacks A or some argument supporting A. But in our case, our notion of defeat (Definition 25) requires that every supporter of A (including A itself) has to be attacked by some argument in $\mathcal{S}$.

In the work of Cayrol; Lagasquie-Schiex (2007), the main idea is to transform a *BAF* into a Dung-like *AAF* that consists of a set of coalitions and an attack relation between them. The support relation is employed to identify coalitions, which are $\subseteq$-maximal conflict-free sets of arguments connected by the support relation. An attack relation is defined over the set of coalitions so that Dung's acceptability semantics can be applied to this coalition framework. According to this approach, in the *BAF* of Figure 15a, there are three coalitions: $\mathcal{C}_1 = \{A\}$, $\mathcal{C}_2 = \{B, C, D\}$ and $\mathcal{C}_3 = \{E, F\}$, where the $\mathcal{C}_1$ attacks $\mathcal{C}_2$ (as $A \in \mathcal{C}_1$ and $B \in \mathcal{C}_2$ and A attacks B) and $\mathcal{C}_2$ attacks $\mathcal{C}_3$ (as $D \in \mathcal{C}_2$ and $E \in \mathcal{C}_3$ and D attacks E). The resulting *AAF* is in Figure 15b, whose unique preferred/stable extension is $\{\mathcal{C}_1, \mathcal{C}_3\}$; the unique corresponding coalition-preferred/coalition-stable extension of $\mathcal{B}$ is given by $\mathcal{C}_1 \cup \mathcal{C}_3 = \{A, E, F\}$. Notice that the attack from A to B was enough to reject the whole coalition $\mathcal{C}_2$ (including the argument D). Besides, as pointed out by Cayrol; Lagasquie-Schiex (2007), some properties of Dung's *AAF*s are not preserved. For instance, the coalition-preferred extension $\{A, E, F\}$ of $\mathcal{B}$ is not admissible in the sense that it cannot defend itself: argument D attacks E, but no argument in

$\{A, E, F\}$ defends E.



(a) *BAF* $\mathcal{B}$ (b) Resulting *AAF*

Figure 15 – *BAF* illustrating the coalition-based semantics from Cayrol; Lagasquie-Schiex (2007).

Our semantics provide a quite distinct result: the unique β-preferred (and β-stable) extension of $\mathcal{B}$ is $\{A, B, C, D, F\}$. A distinguishing aspect of our work, when compared with Cayrol; Lagasquie-Schiex's (2007), is that to defeat an argument A, we have to defeat every supporter of A. In Cayrol; Lagasquie-Schiex's (2007) work, however, a single attack on a supporter of A may be enough to reject the entire coalition with A. In addition, as we can see in Section 3.2, our approach generalizes fundamental properties of Dung's *AAF*s, such as every β-preferred extension of a *BAF* $\mathcal{B}$ being a β-admissible set of $\mathcal{B}$.

Since our approach exclusively relies on deductive support, we will focus on proposals for *BAF*s based on this type of support in the remainder of this section. Boella *et al.* (2010) introduce an approach that enhances the work of Cayrol; Lagasquie-Schiex (2007) by mapping *BAF*s to meta-argumentation frameworks. But instead of grouping arguments in meta-arguments (coalitions) as done by Cayrol; Lagasquie-Schiex (2007), Boella *et al.* (2010) add meta-arguments as follows:

Definition 32 (Boella *et al.* 2010)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. The associated meta-argumentation framework is $\mathcal{EAF} = (\mathcal{MA}, \mathit{Att}')$, where

$$\mathcal{MA} = \{\mathit{acc}(A) \mid A \in \mathcal{A}\} \cup \{X_{A,B}, Y_{A,B}, Z_{A,B} \mid A, B \in \mathcal{A}\},$$

with $X_{A,B}$ read as "*the attack from A to B is not active*"; $Y_{A,B}$, as "*the attack from A to B is active*"; $Z_{A,B}$, as "*A does not support B*"; and $\mathit{acc}(A)$, as "*A is accepted*". The meta-attack relation $\mathit{Att}' \subseteq \mathcal{MA} \times \mathcal{MA}$ satisfies the conditions below:

a) if $(A, B) \in \mathit{Att}$, then $(\mathit{acc}(A), X_{A,B}) \in \mathit{Att}'$, $(X_{A,B}, Y_{A,B}) \in \mathit{Att}'$, and also we have $(Y_{A,B}, \mathit{acc}(B)) \in \mathit{Att}'$;
b) if $(A, B) \in \mathit{Sup}$, then $(\mathit{acc}(B), Z_{A,B}) \in \mathit{Att}'$ and $(Z_{A,B}, \mathit{acc}(A)) \in \mathit{Att}'$.

The extensions of $\mathcal{B}$ are determined by Dung's acceptability semantics for the associated $\mathcal{EAF}$: if $\mathcal{S}$ is a complete, grounded, preferred, stable, or semi-stable extension of $\mathcal{EAF}$, then $\{A \mid acc(A) \in \mathcal{S}\}$ will be respectively a complete, grounded, preferred, stable, or semi-stable extension of $\mathcal{B}$.

In contrast with Cayrol; Lagasquie-Schiex's (2007) approach, Boella *et al.* (2010) develop deductive support and introduce mediated defeats (described below) instead of secondary defeats. For a *BAF* $(\mathcal{A}, \mathit{Att}, \mathit{Sup})$, a *mediated defeat* from A to B is a sequence $(A_1, A_2, \ldots, A_{n-1}, A_n), n \geq 3$, with $A_1 = B, A_n = A$ such that $(A_n, A_{n-1}) \in \mathit{Att}$ and $\forall i \in \{1, \ldots, n-2\}$, it holds $(A_i, A_{i+1}) \in \mathit{Sup}$.

Notice that the explicit inclusion of supported defeats and mediated defeats to a *BAF* does not alter the resulting semantics from Boella *et al.* (2010). This interpretation favors attacks over supports, while ours tends to favor supports over attacks, because to defeat an argument A, we have to defeat every argument supporting A. Hence, in general, neither supported defeats nor mediated defeats are defeats in our sense. Above all, arguments that are not attacked directly will never be defeated in our approach. In the framework of Figure 16, the sequence A supports B, B supports C, and D attacks C is a mediated attack from D to A; it suffices to reject argument A, given D is accepted. Our interpretation admits A and D being both acceptable simultaneously, as no argument directly contests them.

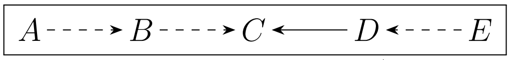


Figure 16 – *BAF* illustrating mediated defeat from D to A.

A meta-argumentation framework is also constructed from coalitions in Cayrol; Lagasquie-Schiex's (2013) approach, but with a notable distinction: arguments are not partitioned into coalitions. Instead, for a *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$, each argument $A \in \mathcal{A}$ gives rise to its coalition $\mathcal{C}(A) = \{B \in \mathcal{A} \mid A \in \mathit{Sup}^*(B)\}$ in the meta-framework defined as the set of arguments directly or indirectly supported by A, where Sup^* is the reflexive and transitive closure of *Sup*. Then, the resulting Dung meta-argumentation framework corresponding to $\mathcal{B}$ is $\mathcal{B}^{\mathcal{C}} = (\mathcal{A}^{\mathcal{C}}, \mathit{Att}^{\mathcal{C}})$, satisfying the conditions below:

a) $\mathcal{A}^{\mathcal{C}} = \{\mathcal{C}(A) \mid A \in \mathcal{A}\}$;
b) $\mathit{Att}^{\mathcal{C}} = \{(\mathcal{C}(A), \mathcal{C}(B)) \mid \exists A' \in \mathcal{C}(A) \textit{ and } \exists B' \in \mathcal{C}(B) \textit{ such that } (A', B') \in \mathit{Att}\}$.

This change enables a one-to-one correspondence between extensions of the meta-framework and the associated Dung framework, with the additional attacks derived from the support relation. The extensions of $\mathcal{B}$ are determined by applying Dung's acceptability semantics to $\mathcal{B}^{\mathcal{C}}$. A set $S \subseteq \mathcal{A}$ is a complete, grounded, preferred, stable, or semi-stable extension of $\mathcal{B}$ iff $\{\mathcal{C}(A) \mid A \in S\}$ is respectively a complete, grounded, preferred, stable, or semi-stable extension of $\mathcal{B}^{\mathcal{C}}$. In the framework of Figure 16, coalitions of A, B, C, D, and E are respectively $\mathcal{C}(A) = \{A, B, C\}$, $\mathcal{C}(B) = \{B, C\}$, $\mathcal{C}(C) = \{C\}$, $\mathcal{C}(D) = \{D\}$, and $\mathcal{C}(E) = \{D, E\}$. It holds $\mathcal{C}(D)$ and $\mathcal{C}(E)$ attack $\mathcal{C}(A), \mathcal{C}(B)$, and $\mathcal{C}(C)$. The unique complete extension of $\mathcal{B}^{\mathcal{C}}$ is $\{\mathcal{C}(D), \mathcal{C}(E)\}$, i.e., the unique complete extension of $\mathcal{B}$ is $\{D, E\}$. Note that the attack from D to C is sufficient to reject A, B, and C, whereas C is not defeated by D in our approach. The unique β-complete extension of $\mathcal{B}$ is $\{A, B, C, D, E\}$. In Cayrol; Lagasquie-Schiex's (2013) work (as in ours), the rejection of an argument supported by A leads to the rejection of A; the difference is the definition of rejection. According to Cayrol; Lagasquie-Schiex (2013), if B is accepted and B attacks A, then A and every argument supporting A is rejected; as for our work, a single attack from an accepted argument is not enough to reject A: in addition, we require that every supporter of A has to be attacked by an accepted argument.

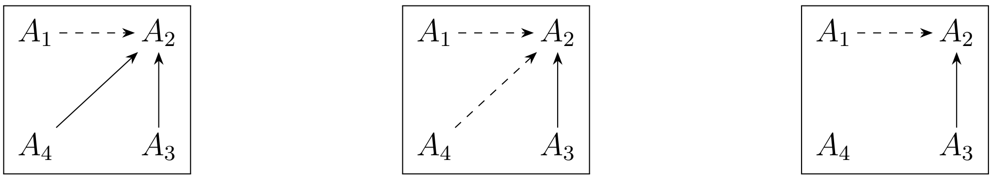
Although the works we have seen so far (Boella *et al.*, 2010; Cayrol; Lagasquie-Schiex, 2005; Cayrol; Lagasquie-Schiex, 2007; Cayrol; Lagasquie-Schiex, 2013) have different points of view, they share a common characteristic: they treat attacks and supports asymmetrically, favoring the attack relation. Once again, regarding the *BAF* of Figure 16, we can observe that in the semantics introduced by Boella *et al.* (2010); Cayrol; Lagasquie-Schiex (2005, 2007, 2013), there is no extension where argument C is acceptable, even though it is both attacked and supported.

Motivated by the deductive supports from Cayrol; Lagasquie-Schiex (2013), Potyka (2020b) proposes *s-deductive* and *m-deductive* labelings to treat attacks and supports equally and generalize Dung's stable semantics from *AAF*s to *BAF*s. However, unlike the stable semantics, *m-deductive* and *s-deductive* semantics permit labeling arguments as undecided in bipolar frameworks. Then, in subsequent work (Potyka, 2021), the same author proposes a generalization of complete (instead of stable) semantics, called bi-complete, to bipolar argumentation frameworks. Semantics as the bi-grounded, bi-preferred, and bi-stable are defined in terms of the bi-complete semantics in the same way that the grounded, preferred, and stable semantics are defined in terms of the complete semantics. In particular, different from *s-deductive* and *m-deductive* semantics, the bi-stable semantics does not allow labeling arguments undecided in

*BAF*s. For a *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$, a labeling $\mathcal{L}$ of $\mathcal{B}$ is bi-complete iff it satisfies the following:

- $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(B) = \texttt{out}$ for every $B \in Att(A)$ or $|\{B \in Att(A) \mid \mathcal{L}(B) \neq \texttt{out}\}| < |\{B \in Sup(A) \mid \mathcal{L}(B) = \texttt{in}\}|$;
- $\mathcal{L}(A) = \texttt{out}$ iff $|\{B \in Sup(A) \mid \mathcal{L}(B) \neq \texttt{out}\}| < |\{B \in Att(A) \mid \mathcal{L}(B) = \texttt{in}\}|$.

A majority voting determines whether attackers or supporters dominate a specific argument. This definition allows for a rejected argument A to be eventually `undec` if enough non-rejected supporters are added, and A could also be eventually accepted if enough accepted supporters are added. Given its treatment of attacks and supports and its generalization of complete semantics to *BAF*s, we can affirm that Potyka's (2021) work is closely related to ours. In common, these works consider that an attack from an accepted argument is not enough to reject an argument. Also, unattacked arguments will always be labeled as `in` in both bi-complete and β-complete labelings. However, they have remarkable differences as we do not consider the number of attackers and supporters to accept an argument; instead, only one accepted supporter of A is enough for A's acceptance.

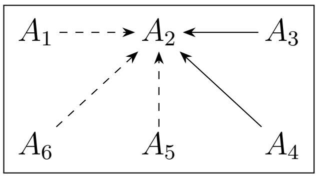


(a) $\mathcal{B}_1$ (b) $\mathcal{B}_2$ (c) $\mathcal{B}_3$

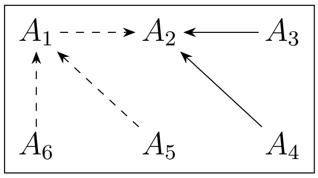
Figure 17 – *BAF*s illustrating the bi-complete semantics from Potyka (2021).

In $\mathcal{B}_1$ of Figure 17a, two arguments are attacking A_2 and one supporting A_2; thus, the attackers prevail, and according to Potyka (2021), the unique bi-complete labeling assigns `out` to A_2 and `in` to the other arguments. On the other hand, in Figure 17b, the supporters of A_2 prevail and the unique bi-complete labeling assigns `in` to A_1, A_2, A_3, and A_4. In Figure 17c, as A_1, A_3, and A_4 have no attackers, they must be labeled `in`; as neither its attackers nor its supporters prevail, A_2 must be labeled `undec` in the unique bi-complete labeling of $\mathcal{B}_3$. In each of these frameworks, argument A_2 has a distinct label in the corresponding bi-complete labeling; depending on the balance between its attackers and supporters, A_2 can be `in`, `out`, or `undec`. In our work, the support of the unattacked argument A_1 guarantees that A_2 is `in` in the corresponding β-complete labeling; then $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ have the same β-complete labeling assigning `in` to each argument, including A_2.

One weakness of counting votes from arguments is that an agent can provide many



(a) $\mathcal{B}_1$



(b) $\mathcal{B}_2$

Figure 18 – *BAF*s illustrating how the bi-complete semantics differ from ours in handling indirect supporters.

similar or even redundant arguments to support or attack a specific argument, which could lead to an unfair evaluation. Besides, numeric majority is not always adequate to encode the strength of arguments. Recall Example 8 (Tweety example), where we wish to encode an exception to a general rule. Using the bi-complete interpretation of support, the framework for this scenario can contain both attacks and supports (as in our approach using the β-semantics), because a conflict between arguments can be eventually lifted with enough supporters. However, in the general case, it would require a possibly large (and variable) number of supporting arguments (depending on the number of attacking arguments), hindering its use to represent exceptions to a general rule.

Another distinction is that the characterization of the bi-complete labelings does not consider the indirect supporters of an argument. In this case, in $\mathcal{B}_1$ of Figure 18a, two arguments are attacking and three are supporting A_2; thus, the supporters prevail, and according to Potyka (2021), the unique bi-complete labeling assigns `in` to every argument in the *BAF*, including A_2. On the other hand, in $\mathcal{B}_2$ of Figure 18b, the attackers of A_2 prevail (as the only direct supporter A_1 is taken into account) and the unique bi-complete labeling assigns `out` to A_2 and `in` to the remaining arguments. In our work, $\mathcal{B}_1$ and $\mathcal{B}_2$ have the same β-complete labeling that assigns `in` to each argument.

In the next section, we summarize the key findings of this chapter and contextualize them to the general objectives of our work.

3.6 Conclusions

In this chapter, we have introduced new semantics for Bipolar Argumentation Frameworks (*BAF*s). These semantics, namely β-admissible, β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable, generalize the corresponding semantics from Dung's (1995) framework to *BAF*s. We have presented both extension-based and labeling-based formulations for them.

A key concept in our proposal is the notion of defeat, where an argument A is defeated by a set of arguments $\mathcal{S}$ if every (direct or indirect) supporter of A is attacked by at least one argument in $\mathcal{S}$. We then offer a dual treatment for defeat and defense. Specifically, $\mathcal{S}$ defeats A iff $\mathcal{S}$ defeats every supporter of A (where we consider A as a supporter of itself). Similarly, $\mathcal{S}$ defends A iff $\mathcal{S}$ defends every argument supported by A. Roughly speaking, we can interpret here the defeat of an argument by a set $\mathcal{S}$ of arguments as a successful collective attack from $\mathcal{S}$ in the same way as we can interpret its defense as a successful collective support from $\mathcal{S}$.

These notions of defeat and defense have been designed to collapse into the corresponding concepts proposed for *AAF*s by Dung (1995) when the support relation is ignored. Building upon these dual notions of defeat and defense, we define conflict-free sets, acceptability, extension-based, and labeling-based semantics. In *AAF*s, a conflict arises from attacks; here it arises from defeats, i.e., a conflict-free set $\mathcal{S}$ in our conception is a set that does not defeat any of its elements. Acceptable arguments with respect to a set are defined as in *AAF*s, but resorting to our definition of defense. Then, we characterize our semantics by mimicking the extension-based semantics defined by Dung (1995) and the labeling-based semantics defined by Caminada (2006a) for *AAF*s. Besides generalizing *AAF* semantics to *BAF*s, our approach also preserves some of their most remarkable results. In particular, we have generalized Dung's (1995) Fundamental Lemma to *BAF*s. This lemma plays a crucial role in our semantics, providing a solid foundation for reasoning about argument acceptability in our framework.

As the combination of support and attack is controversial (even in simple frameworks), there are several proposals with different intuitions and motivations. We do not claim our approach is better than another one, but that it is worthwhile due to the following:

a) it provides a comprehensive framework for evaluating arguments in a bipolar context;
b) naturally generalizes Dung's approach;
c) preserves fundamental results of *AAF*s.

In this sense, it opens up new opportunities for studying and analyzing argumentation in various domains, contributing to the advancement of the field of Argumentation Theory. In the next chapter, we compare our *BAF* semantics with existing logic programming semantics to gain further insights into their relationships and distinctive features.

4 ON THE EQUIVALENCE BETWEEN LOGIC PROGRAMS AND BIPOLAR ARGUMENTATION FRAMEWORKS

In this chapter, we leverage the β-semantics defined in the previous chapter to elicit connections between Bipolar Argumentation Frameworks (BAFs) and Normal Logic Programs (NLPs) by providing direct translations from BAFs to NLPs (and vice versa) in a one-to-one correspondence between the β-semantics and logic programming semantics. This includes the equivalence involving the L-stable semantics. Besides, we also obtain syntactic equivalences between NLPs and BAFs by finding subsets of them for which the proposed translations are each other's inverse up to isomorphism.

Parts of this chapter appeared in the work of Alcântara; Cordeiro (2025).

4.1 Introduction

There are already several links between *BAF*s (Definition 15) and *NLP*s (Definition 19), though not for all semantics or not in both directions:

a) Čyras; Schulz; Toni (2017) translate *BAF*s into Assumption-Based Argumentation (*ABA*) frameworks for four interpretations of support (Nouioua; Risch, 2011; Cayrol; Lagasquie-Schiex, 2013; Cayrol; Lagasquie-Schiex, 2013; Gabbay, 2016), while preserving the admissible, preferred, and stable semantics. As there is a semantic correspondence between *ABA* frameworks and *NLP*s in both directions (Caminada; Schulz, 2017), we can try to link *BAF*s and *NLP*s for the admissible, preferred, and stable semantics in one direction: any *BAF* can be translated to an *ABA* framework, which in turn corresponds to some *NLP*. However, the inverse direction from *NLP*s to *BAF*s and the correspondence between the semi-stable and L-stable semantics remain unclear, as Čyras; Schulz; Toni (2017) study only the direction from *BAF*s to *ABA* frameworks and consider only the admissible, preferred, and stable semantics;

b) rather than composing transformations, Nouioua; Boutouhami (2023) present a direct link between *BAF*s and *NLP*s by providing a one-to-one semantic correspondence from *BAF*s to *NLP*s, and vice versa, between the main acceptability semantics and logic programming semantics, with a notable exception: in general, the L-stable semantics of an *NLP* does not correspond to the semi-stable semantics of its corresponding *BAF*.

c) in the context of recursive argumentation, where attacks/supports can target attacks/supports, Alfano *et al.* (2024) map recursive *BAF*s (Rec-*BAF*) to *NLP*s, with complete extensions corresponding to partial stable models. They also consider non-recursive *BAF*s, but do not study in neither case the semi-stable/L-stable semantics nor the inverse direction: as any *AAF* is a (recursive) *BAF*, Caminada; Sá, *et al.*'s (2015) translation from *NLP*s to *AAF*s is already sufficient for semantic correspondence from *NLP*s to (recursive) *BAF*s when not considering the semi-stable/L-stable semantics.

In summary, the semi-stable/L-stable semantics remain a primary gap in the connections between *NLP* semantics and these semantics for *BAF*. Additionally, these works have not explored whether it is possible to find a link between *BAF*s and *NLP*s such that the translation from *BAF*s to *NLP*s acts as an inverse of the translation from *NLP*s to *BAF*s (when ignoring names of arguments/atoms). This motivates our focus on the following questions:

a) whether *NLP*s can be translated to *BAF*s with a one-to-one semantic correspondence (including one semantics for *BAF* corresponding to L-stable);
b) to what extent the translations from *NLP*s to *BAF*s and from *BAF*s to *NLP*s are the inverse of each other.

We leverage the remarkable characteristics of the β-semantics introduced in Chapter 3, namely its dual criteria for acceptance and rejection of arguments. An argument A is accepted by a set of arguments $\mathcal{S}$ if at least a direct or indirect supporter of A (notice A is a supporter of itself) is not attacked by any argument in $\mathcal{S}$; on the other hand, A is rejected by $\mathcal{S}$ if every direct or indirect supporter of A is attacked by at least one argument in $\mathcal{S}$. Taking profit from this dual perspective, we establish in this chapter a one-to-one correspondence between β-semantics and various logic programming semantics, including the challenging L-stable semantics. We recall that for traditional attack-only *AAF*s, the translation proposed by Caminada; Sá, *et al.* (2015) from *NLP*s to *AAF*s does not guarantee semantic correspondence between (argumentation) semi-stable and (logic programming) L-stable semantics.

To prove this semantic equivalence for *BAF*s, we provide direct translations between *BAF*s and *NLP*s (and their semantics) in both directions. We also show that the relationship between *NLP*s and *BAF*s is demonstrably more robust than the one between *NLP*s and *AAF*s, extending beyond semantics to encompass structural aspects. For example, we find classes of *BAF*s and *NLP*s, respectively redundancy-free *BAF*s of support cliques ($\mathfrak{S}$-*RFBAF*s) and Reduced Atomic Logic Programs (*RALP*s), for which these transformations act as inverses (up to isomorphism) of each other, meaning that each transformation undoes the other if we ignore the names of arguments and atoms. This elicits a notion of structural (or syntactic) equivalence, as no relevant information is lost by translating a formalism into the other. In addition, we find that the class of *NLP*s that are structurally equivalent to *BAF*s is as expressive as the set of all *NLP*s, meaning that any *NLP* is semantically equivalent to (i.e., as expressive as) the corresponding *NLP* of some *BAF*.

As a collateral effect of this semantic and structural equivalence, our results reveal that the β-semantics interpretation of support aligns with the *NLP* semantics, offering new insights for knowledge representation and reasoning with *BAF*s.

Several works also relate more expressive argumentation formalisms to logic programming. For instance, Abstract Dialectical Frameworks (*ADF*s) (Brewka; Woltran, 2010; Strass, 2013; Alcântara; Sá; Acosta-Guadarrama, 2019), Assumption-Based Argumentation (*ABA*) (Caminada; Schulz, 2017), Claim-augmented Argumentation Frameworks (*CAF*s) (Rocha; Cozman, 2022; König; Rapberger; Ulbricht, 2022), Frameworks with Sets of Attacking Arguments (*SETAF*s) (König; Rapberger; Ulbricht, 2022) and many others (Alfano *et al.*, 2020) are connected to logic programming. We list some of these approaches below:

a) the semi-stable semantics of an *ABA* framework and the L-stable semantics of its corresponding *NLP* do not coincide according to Caminada; Schulz's (2017) translation of *ABA* frameworks into *NLP*s;
b) Brewka; Woltran's (2010); Strass's (2013) approaches fail to express in *ADF*s the partial stable semantics of *NLP*s. Although this is enabled by Alcântara; Sá; Acosta-Guadarrama's (2019) approach, it lacks a structural correspondence, as the translations from *ADF*s to *NLP*s, and vice versa, are not each other’s inverse, not even up to isomorphism;
c) the work most closely related to ours is that of Rocha; Cozman (2022), where Bipolar Claim-augmented Argumentation Frameworks (*BCAF*s) are related to *NLP*s. The authors establish a one-to-one correspondence between *BCAF*s and *NLP*s in both directions, while preserving the main semantics, including the L-

stable semantics. The main distinction between *BCAF*s and *BAF*s (and hence between their work and ours) lies in the additional structure of *BCAF*s: each argument in a *BCAF* is associated with an explicit claim (or conclusion). In contrast, arguments in a *BAF* are completely abstract, with the β-semantics being dependent exclusively on the attack and support relation between arguments, treated as objects without any internal structure. In this sense, to the best of our knowledge, ours is the first work to provide translations from *BAF*s to *NLP*s, and vice versa, preserving the semantics based on partial stable models, including the L-stable semantics. Unlike Rocha; Cozman's (2022) work, we also find classes of *NLP*s and *BAF*s for which the formalisms are structurally equivalent.

This chapter is organized as follows: in Section 4.2, we provide a translation from *NLP*s to *BAF*s and between their semantics; the inverse direction from *BAF*s to *NLP*s is discussed in Section 4.3; in the following, we introduce in Section 4.4 the classes of Reduced Atomic Logic Programs (*RALP*s) and redundancy-free *BAF*s of support cliques ($\mathfrak{S}$-*RFBAF*) for which the proposed translations act as each other's inverse up to isomorphism; we relate *NLP*s to *RALP*s in Section 4.5 and show they are equally expressive for the semantics considered in our work; we collect our conclusions for the chapter in Section 4.6.

4.2 From *NLP*s to *BAF*s

In this section, we revisit the three-step process of establishing the argumentation framework used by Caminada; Sá, *et al.* (2015) to translate a *NLP* into an *AAF*. This method is based on the approach introduced by Wu; Caminada; Gabbay (2009) and shares similarities with the procedures used in ASPIC (Caminada; Amgoud, 2005; Caminada; Amgoud, 2007) and logic-based argumentation (Gorogiannis; Hunter, 2011). Its first step involves taking an *NLP* and constructing its corresponding *AAF*. Then we apply *AAF* semantics in the second step, followed by an analysis of the implications of these semantics at the level of conclusions (step 3). The novelty of our approach is that we include a support relation between arguments with the same conclusion in the first step. We now detail this process.

4.2.1 BAF construction

We will present a translation from *NLP*s to *BAF*s sufficiently robust to establish a correspondence between partial stable models and β-complete labelings, well-founded models and β-grounded labelings, regular models and β-preferred labelings, stable models and β-stable labelings, L-stable models and β-semi-stable labelings.

When translating *NLP*s into *BAF*s, each argument in the corresponding *BAF* represents a derivation for an atom of the original *NLP*.

Definition 33 (Caminada; Sá, *et al.* 2015)**.** From an *NLP* P, we can start to construct arguments recursively as follows:

a) if $c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is a rule in P, it is also an argument (say A) with the following components:
   - $\texttt{Conc}(A) = c$;
   - $\texttt{Rules}(A) = \{c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$;
   - $\texttt{Vul}(A) = \{b_1, \ldots, b_m\}$;
   - $\texttt{Sub}(A) = \{A\}$.

b) if $c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is a rule (say r) in P and for each a_i $(1 \leq i \leq n)$ there exists an argument A_i with $\texttt{Conc}(A_i) = a_i$ and $c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is not contained in $\texttt{Rules}(A_i)$, then r derives the argument $c \leftarrow (A_1), \ldots, (A_n), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ (say A) with the following components:
   - $\texttt{Conc}(A) = c$;
   - $\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{r\}$;
   - $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \ldots \cup \texttt{Vul}(A_n) \cup \{b_1, \ldots, b_m\}$;
   - $\texttt{Sub}(A) = \{A\} \cup \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n)$.

By $\mathcal{A}_P$ we mean the set of all arguments constructed from P as above. Each argument A represents a derivation in P for the atom $\texttt{Conc}(A)$. Below, we explain the intuition behind some of the definitions above:

a) $\texttt{Rules}(A)$ is the set of all rules used for deriving $\texttt{Conc}(A)$. For instance, for a program with three rules $r_1 = c \leftarrow a, \texttt{not}\ b$ and $r_2 = a \leftarrow \texttt{not}\ a$ and $r_3 = a \leftarrow \texttt{not}\ c$, we find an argument C_1 with $\texttt{Conc}(C_1) = c$ and $\texttt{Rules}(C_1) = \{r_1, r_2\}$, and also an argument C_2 with $\texttt{Conc}(C_2) = c$ and $\texttt{Rules}(C_2) = \{r_1, r_3\}$. The latter used r_3 instead of r_2 for deriving c. The main purpose of $\texttt{Rules}(\cdot)$ is keeping track of which rules were already used, so that no argument can use a same rule twice for deriving some conclusion. For example, for a program with two rules $r'_1 = c \leftarrow a, \texttt{not}\ b$ and $r'_2 = a \leftarrow a, \texttt{not}\ d$, we cannot build an argument A with $\texttt{Conc}(A) = a$, as using the rule $a \leftarrow a, \texttt{not}\ d$ requires the use of some *other* rule for a;

b) $\texttt{Vul}(A)$ is the set of vulnerabilities of the argument/derivation A. In any partial stable model $\mathcal{M}$ of P, a rule r is effectively ignored if there is some $v \in \mathbf{t}(\mathcal{M})$ such that $\texttt{not}\ v \in \textit{body}^-(r)$. Thus, a vulnerability of A is an atom that appears in the negative body of some rule in the derivation. Vulnerabilities of an argument A will be used to determine the attackers of A;

c) $\texttt{Sub}(A)$ is the set of subarguments/subderivations used in an argument/derivation A. For each positive body atom a in a rule deriving argument C with $\texttt{Conc}(C) = c$, we require some argument A with $\texttt{Conc}(A) = a$ to be a subargument of C. Stated differently, if viewing a derivation A as a tree, the (proper) subarguments of A would be the (proper) subtrees of A. While the notion of subarguments is not used for defining corresponding *BAF*s, it is a useful concept when proving some theorems or talking about derivations.

Now we will clarify the connection between the existence of arguments and the existence of a derivation in a reduct (recall Definition 21).

Lemma 34. Let P be an *NLP*, $\mathcal{I}$ an interpretation, and $\Psi^{\uparrow\omega}_{P/\mathcal{I}}$ the least 3-valued model of $P/\mathcal{I}$ (see Definition 22). The following holds:

a) $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$;

b) $c \in \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff for each argument A constructed from P such that $\texttt{Conc}(A) = c$, we have $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) \neq \emptyset$.

Lemma 34 ensures that arguments are closely related to derivations in a reduct. An atom c is true in the least 3-valued model of $P/\mathcal{I}$ iff we can construct an argument with conclusion c and whose vulnerabilities are false according to $\mathcal{I}$; otherwise, c is false in the least 3-valued model of $P/\mathcal{I}$ iff for every argument whose conclusion is c, at least one of its vulnerabilities is true in $\mathcal{I}$. The next result is a direct consequence of Lemma 34:

Corollary 35. Let P be an *NLP* and $c \in \textit{HB}_P$. The two statements below hold:

a) $c \in \mathbf{t}(\Psi^{\uparrow \omega}_{P/\mathcal{I}})$ for $\mathcal{I} = (\emptyset, \mathit{HB}_P, \emptyset)$ iff there exists an argument A constructed from P such that $\mathtt{Conc}(A) = c$;

b) $c \in \mathbf{f}(\Psi^{\uparrow \omega}_{P/\mathcal{I}})$ for every interpretation $\mathcal{I}$ iff there is no argument A constructed from P such that $\mathtt{Conc}(A) = c$.

Considering Lemma 34 for $\mathcal{I} = (\emptyset, \mathit{HB}_P, \emptyset)$, we conclude that there is an argument A from P with conclusion c iff c is true in the least 3-valued model of $P/\mathcal{I}$, because the condition $\mathtt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$ becomes trivially true for $\mathbf{f}(\mathcal{I}) = \mathit{HB}_P$. On the other hand, the atoms that are lost in translation (i.e., the atoms not associated with arguments) are simply those that are false in the least 3-valued model of every possible reduct of P.

An argument can be seen as a tree-like structure representing a possible derivation of an atom from the rules of a program.

Example 13. Consider the *NLP* P below with rules $\{r_1, \ldots, r_8\}$:

$r_1 : \; a$	$r_2 : \; b \leftarrow a$	$r_3 : \; c \leftarrow \mathtt{not}\ c$
$r_4 : \; d \leftarrow b, \mathtt{not}\ a$	$r_5 : \; d \leftarrow \mathtt{not}\ c$	$r_6 : \; e \leftarrow b, c, \mathtt{not}\ e$
$r_7 : \; c \leftarrow f, \mathtt{not}\ g$	$r_8 : \; f \leftarrow c, g.$	

According to Definition 33, we can construct the following arguments from P:

$A_1 : \; a$	$A_2 : \; b \leftarrow (A_1)$	$A_3 : \; c \leftarrow \mathtt{not}\ c$
$A_4 : \; d \leftarrow (A_2), \mathtt{not}\ a$	$A_5 : \; d \leftarrow \mathtt{not}\ c$	$A_6 : \; e \leftarrow (A_2), (A_3), \mathtt{not}\ e$

In Table 2, we give the conclusions and vulnerabilities of each argument. Alternatively, we can depict arguments as possible derivations as in Figure 19.

	A_1	A_2	A_3	A_4	A_5	A_6
$\mathtt{Conc}(.)$	a	b	c	d	d	e
$\mathtt{Vul}(.)$	$\emptyset$	$\emptyset$	$\{c\}$	$\{a\}$	$\{c\}$	$\{c, e\}$

Table 2 – Conclusions and vulnerabilities of arguments from Example 13.

The vulnerabilities of an argument A are associated with the negative literals found in the derivation of A. If $\mathtt{not}\ a$ is one of them, we know that a is one of its vulnerabilities. This means that if a is derived, then $\mathtt{Conc}(A)$ cannot be obtained via this derivation represented by A. However, it can still be obtained via other derivations/arguments. For instance, in the program P of Example 13, the derivation of a suffices to prevent the derivation of d via argument A_4 (for that reason, $a \in \mathtt{Vul}(A_4)$), but we still can derive d via A_5. Notice also that there are no

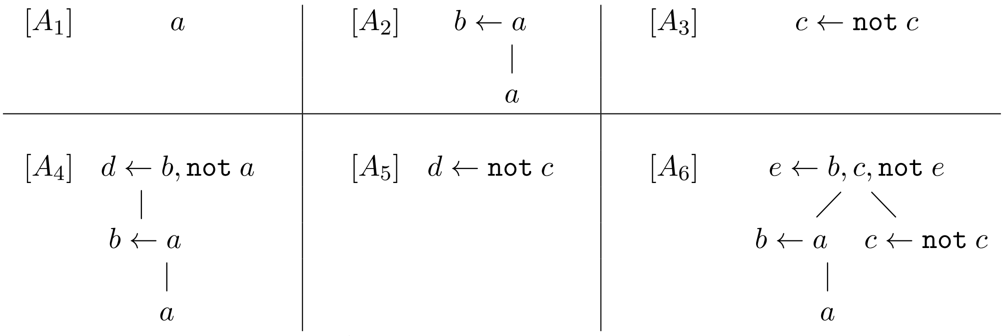


Figure 19 – Arguments constructed from P.

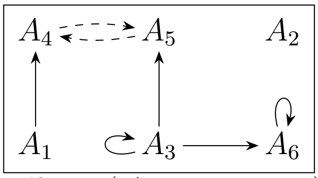


Figure 20 – The corresponding *BAF* $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$.

arguments with conclusions f and g. From Corollary 35, we know that it is not possible to derive them from P as they are false in the least three-valued model of each reduct of P.

Now we are entitled to determine both the attack and support relations:

Definition 34. Let P be an *NLP* and let A and B be arguments in the sense of Definition 33. We define the attack relation Att_P and the support relation Sup_P as follows:

a) $(A, B) \in Att_P$ iff $\texttt{Conc}(A) \in \texttt{Vul}(B)$;
b) $(A, B) \in Sup_P$ iff $A \neq B$ and $\texttt{Conc}(A) = \texttt{Conc}(B)$.

Our definition of the set Att_P of attacks obtained from an *NLP* P is exactly as it has been characterized by Caminada; Sá, *et al.* (2015). The novelty here is the definition of Sup_P, which in our proposal is determined by arguments with the same conclusion (the purpose of the condition $A \neq B$ is to avoid incorporating the redundant case of an argument supporting itself). For the arguments of Example 13, A_1 attacks A_4; A_3 attacks itself, A_5, and A_6; A_6 attacks itself. We also have A_4 and A_5 supporting each other. Using the thus-defined concepts of arguments, attacks and supports, we can define the corresponding *BAF* of a particular *NLP*:

Definition 35 (Corresponding *BAF*)**.** Let P be an *NLP*. We define its corresponding *BAF* as $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$, where $\mathcal{A}_P$ is the set of arguments in the sense of Definition 33, and Att_P and Sup_P are respectively the attack and support relations in the sense of Definition 34.

As an example, the corresponding *BAF* $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ of the *NLP* of Example 13 is depicted in Figure 20.

4.2.2 Equivalence results

In the sequel, we show the equivalence between the semantics of an *NLP* P and their counterpart for the corresponding *BAF* $\mathcal{B}_P$. For this purpose, we introduce two functions: $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}$, which maps each labeling to an interpretation, and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}$, which maps each interpretation to a labeling. We then investigate the conditions under which these functions act as inverses of each other and employ these results to prove the equivalence between the semantics. Notably, these functions permit us to treat interpretations and labelings interchangeably.

Definition 36 ($\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}$ and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}$ Functions)**.** Let P be an *NLP*, $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*, $\mathcal{I}nt$ be the set of all the 3-valued interpretations of P, and $\mathcal{L}ab$ be the set of all labelings of $\mathcal{B}_P$. We introduce a function $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P} : \mathcal{L}ab \to \mathcal{I}nt$ such that $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}) = (T, F, U)$, in which T, F, and U are defined as follows:

a) $T = \{c \in HB_P \mid \mathcal{L}(A) = \texttt{in}$ *for some* $A \in \mathcal{A}_P$ *such that* $\texttt{Conc}(A) = c\}$;

b) $F = \{c \in HB_P \mid \mathcal{L}(A) = \texttt{out}$ *for every* $A \in \mathcal{A}_P$ *such that* $\texttt{Conc}(A) = c\}$;

c) $U = \{c \in HB_P \mid c \notin T \cup F\}$.

We introduce a function $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P} : \mathcal{I}nt \to \mathcal{L}ab$ such that for any $\mathcal{I} \in \mathcal{I}nt$ and any $A \in \mathcal{A}_P$, the following conditions hold:

a) $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{I})(A) = \texttt{in}$ if $\mathcal{I}(\texttt{Conc}(A)) = \mathbf{t}$;

b) $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{I})(A) = \texttt{out}$ if $\mathcal{I}(\texttt{Conc}(A)) = \mathbf{f}$;

c) $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{I})(A) = \texttt{undec}$ if $\mathcal{I}(\texttt{Conc}(A)) = \mathbf{u}$.

The direction from interpretations to labelings is straightforward: the label `in`, `out`, or `undec` assigned to an argument A follows respectively from the truth-value **t**, **f**, or **u** of its conclusion $\texttt{Conc}(A)$. Notably, arguments with the same conclusion will always have the same label. From labelings to interpretations, the connection is clear for atoms $c \in HB_P$ with some argument A such that $\texttt{Conc}(A) = c$. In this case, c is interpreted as true if some argument A with $\texttt{Conc}(A) = c$ is accepted; c is interpreted as false if every argument A with $\texttt{Conc}(A) = c$ is rejected; otherwise, c is undecided. For any atom c with no corresponding argument, c is interpreted as false.

In general, $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))$ is not equal to $\mathcal{L}$. For instance, considering the labeling $\mathcal{L} = (\emptyset, \{A_4\}, \{A_1, A_2, A_3, A_5, A_6\})$ and the *BAF* $\mathcal{B}_P$ of Figure 20, we have that $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) = (\emptyset, \emptyset, \{A_1, A_2, A_3, A_4, A_5, A_6\})$. Such an inequality will always occur when two arguments with the same conclusion receive distinct labels. In our example,

A_4 and A_5 have the same conclusion d, but $\mathcal{L}(A_4) = \texttt{out}$ and $\mathcal{L}(A_5) = \texttt{undec}$. It holds $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})(d) = \mathbf{u}$, and consequently,

$$\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))(A_4) = \mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))(A_5) = \texttt{undec},$$

i.e., $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))$ is not equal to $\mathcal{L}$. However, if $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, arguments with the same conclusion will always have the same label:

Theorem 36. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$ and $\texttt{Conc}(A) = \texttt{Conc}(B)$, then $\mathcal{L}(A) = \mathcal{L}(B)$.

Thus, for β-complete labelings, we have that $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) = \mathcal{L}$:

Theorem 37. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. For any β-complete labeling $\mathcal{L}$ of $\mathcal{B}_P$, it holds $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) = \mathcal{L}$.

In general, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{I}))$ is not equal to $\mathcal{I}$, because of those atoms c occurring in an *NLP* P, but which are not the conclusion of any argument in $\mathcal{A}_P$. However, when $\mathcal{M}$ is a partial stable model, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})) = \mathcal{M}$:

Theorem 38. Let P be an *NLP*, $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P)$ be its corresponding *BAF*, and $\mathcal{M}$ be a partial stable model of P. It holds that $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})) = \mathcal{M}$.

This means that when restricted to β-complete labelings and partial stable models, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}$ and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}$ are each other's inverse. The function $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}$ when applied to β-complete labelings can be characterized as follows:

Proposition 39. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, then

$$\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})(c) = \begin{cases} \mathbf{t} & \exists A \in \mathcal{A}_P \textit{ such that } \texttt{Conc}(A) = c \textit{ and } \mathcal{L}(A) = \texttt{in} \\ \mathbf{u} & \exists A \in \mathcal{A}_P \textit{ such that } \texttt{Conc}(A) = c \textit{ and } \mathcal{L}(A) = \texttt{undec} \\ \mathbf{f} & \textit{otherwise.} \end{cases}$$

In β-complete labelings, different arguments with the same conclusion support each other and share the same acceptability degree. Hence, the acceptability of an atom c follows directly from the acceptability of arguments with conclusion c. If those arguments are accepted (resp. rejected, undecided), c is interpreted as true (resp. false, undecided). If there are no arguments with conclusion c, then c is interpreted as false.

From Lemma 34, Theorems 37 and 38, we can obtain the following result:

Theorem 40. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P;

b) $\mathcal{M}$ is a partial stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P$.

Theorem 40 is one of the main results of this paper. It plays a central role in ensuring the equivalence between the semantics for *NLP*s and their counterpart for *BAF*s:

Theorem 41. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a well-founded model of P;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a regular model of P;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a stable model of P;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is an L-stable model of P.

The following result is a direct consequence of Theorems 38 and 41:

Corollary 42. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{M}$ is a well-founded model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-grounded labeling of $\mathcal{B}_P$;

b) $\mathcal{M}$ is a regular model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-preferred labeling of $\mathcal{B}_P$;

c) $\mathcal{M}$ is a stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-stable labeling of $\mathcal{B}_P$;

d) $\mathcal{M}$ is an L-stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-semi-stable labeling of $\mathcal{B}_P$.

Next, we consider the *NLP* exploited by Caminada; Sá, *et al.* (2015) as a counterexample to show that in general, L-stable models and semi-stable labelings do not coincide with each other in their translation from *NLP*s to *AAF*s:

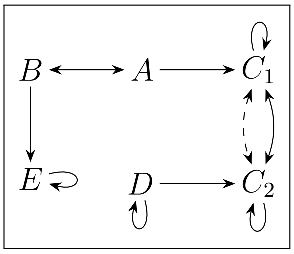
Example 14. Let P be the *NLP* depicted in Figure 21 and $\mathcal{B}_P$ be its corresponding *BAF*, with conclusions and vulnerabilities given in Table 3.

	A	B	C_1	C_2	D	E
`Conc(.)`	a	b	c	c	d	e
`Vul(.)`	$\{a\}$	$\{b\}$	$\{a, c\}$	$\{c, d\}$	$\{d\}$	$\{b, e\}$

Table 3 – Conclusions and vulnerabilities of arguments from Example 14.

$c \leftarrow \texttt{not}\ c, \texttt{not}\ d$	$d \leftarrow \texttt{not}\ d$
$c \leftarrow \texttt{not}\ c, \texttt{not}\ a$	$b \leftarrow \texttt{not}\ a$
$e \leftarrow \texttt{not}\ e, \texttt{not}\ b$	$a \leftarrow \texttt{not}\ b$

(a) P



(b) $\mathcal{B}_P$

Figure 21 – *NLP* P and its associated *BAF* $\mathcal{B}_P$.

Partial Stable Models	Complete labelings
$\mathcal{M}_1 = (\emptyset, \emptyset, HB_P)$	$\mathcal{L}_1 = (\emptyset, \emptyset, \{A, B, C_1, C_2, D, E\})$
$\mathcal{M}_2 = (\{a\}, \{b\}, \{c, d, e\})$	$\mathcal{L}_2 = (\{A\}, \{B\}, \{C_1, C_2, D, E\})$
$\mathcal{M}_3 = (\{b\}, \{a, e\}, \{c, d\})$	$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$
Well-Founded Model	Grounded labeling
$\mathcal{M}_1 = (\emptyset, \emptyset, HB_P)$	$\mathcal{L}_1 = (\emptyset, \emptyset, \{A, B, C_1, C_2, D, E\})$
Regular Models	Preferred labelings
$\mathcal{M}_2 = (\{a\}, \{b\}, \{c, d, e\})$	$\mathcal{L}_2 = (\{A\}, \{B\}, \{C_1, C_2, D, E\})$
$\mathcal{M}_3 = (\{b\}, \{a, e\}, \{c, d\})$	$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$
Stable Models	Stable labelings
None	None
L-stable Models	Semi-stable labelings
$\mathcal{M}_3 = (\{b\}, \{a, e\}, \{c, d\})$	$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$

Table 4 – Semantics for P and $\mathcal{B}_P$.

Note that in $\mathcal{B}_P$ there is both a mutual attack and a mutual support between arguments C_1 and C_2. As expected from Theorems 40 and 41, we obtain in Table 4 the equivalence between partial stable models and β-complete labelings, well-founded models and β-grounded labelings, regular models and β-preferred labelings, stable models and β-stable labelings, L-stable models and β-semi-stable labelings. We emphasize the coincidence involving L-stable models in Table 4 as it does not occur in Caminada; Sá, *et al.*'s (2015) approach. In that reference, the corresponding *AAF* has two semi-stable labelings in contrast to the unique L-stable model $\mathcal{M}_3$ of P.

Note that if the support relation was ignored in the framework of Figure 20 as was done by Caminada; Sá, *et al.* (2015), the β-complete labelings of $\mathcal{B}_P$ would be $\mathcal{L}_1, \mathcal{L}'_2$, and $\mathcal{L}_3$, where

$$\mathcal{L}'_2 = (\{A\}, \{B, C_1\}, \{C_2, D, E\}),$$

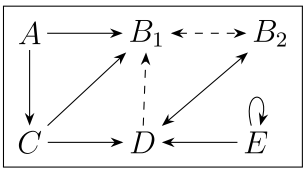
and, as $\mathcal{L}'_2$ and $\mathcal{L}_3$ would be the β-semi-stable labelings, the correspondence with L-stable models would not sustain anymore.

4.3 From *BAF*s to *NLP*s

Now we will provide a translation for the opposite direction, i.e., from *BAF*s to *NLP*s. As in the previous section, this translation guarantees the equivalence between the semantics for *NLP*s and their counterpart for *BAF*s. As we are interested in syntatic correspondences, this translation should also be the inverse of the one proposed in Section 4.2, at least when ignoring argument/atom names.

Before proceeding to the actual translation, we illustrate why a simple translation based on that of Caminada; Sá, *et al.* (2015) is not sufficient for establishing syntactic correspondences. In Caminada; Sá, *et al.*'s (2015) translation, each argument A originates a rule r with $head(r) = A$, $body^+(r) = \emptyset$, and $body^-(r) = \{\texttt{not}\ B \mid B \in Att(A)\}$. For instance, we obtain the rule $A \leftarrow \texttt{not}\ B, \texttt{not}\ C$ from an argument A attacked by (only) B and C. In the following example, we adapt this idea to the context of *BAF*s:

Example 15. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be the *BAF* of Figure 22a. The program P in Figure 22b is very similar to that obtained from Caminada; Sá, *et al.* (2015): the difference is that for each support from X to Y we add the rule $Y \leftarrow X$. In particular, for mutually supporting arguments B_1 and B_2, we add both $B_1 \leftarrow B_2$ and $B_2 \leftarrow B_1$.



(a) $\mathcal{B}$

A
$B_1 \leftarrow \texttt{not}\ A,\ \texttt{not}\ C$
$B_2 \leftarrow \texttt{not}\ D$
$C \leftarrow \texttt{not}\ A$
$D \leftarrow \texttt{not}\ B_2,\ \texttt{not}\ C,\ \texttt{not}\ E$
$E \leftarrow \texttt{not}\ E$
$B_1 \leftarrow D$
$B_1 \leftarrow B_2$
$B_2 \leftarrow B_1$

(b) *NLP* P

Figure 22 – (a) *BAF* $\mathcal{B}$ and (b) *NLP* P from Example 15.

While it is capable of preserving semantics, this simple translation does not suffice for finding syntactic correspondences. The reason is twofold:

a) the translation from *NLP*s to *BAF*s (Section 4.2) includes mutual support between arguments with the same conclusion (i.e., arguments that are derived from rules with the same head), while in the direction from *BAF*s to *NLP*s as described by Example

15, mutual support between arguments is represented by rules with distinct heads. For instance, B_1 and B_2 in Figure 22 mutually support each other, yet they derive rules $B_1 \leftarrow B_2$ and $B_2 \leftarrow B_1$ (with $B_1 \neq B_2$) for encoding their support interaction;

b) the translation from *NLP*s to *BAF*s (Section 4.2), when applied to P from Figure 22b, can generate an argument X from $B_2 \leftarrow \texttt{not}\ D$ and also an argument Y from the rules $B_2 \leftarrow B_1$ and $B_1 \leftarrow \texttt{not}\ A, \texttt{not}\ C$. Both X and Y have conclusion B_2, which means they both attack the argument Z derived from the rule $D \leftarrow \texttt{not}\ B_2, \texttt{not}\ C, \texttt{not}\ E$. Intuitively, X and Y represent ways of deriving the atom B_2, and Z represents the only way to derive the atom D. Also notice that atoms B_2 and D come from the *BAF* $\mathcal{B}$ (Figure 22a). In $\mathcal{B}$, argument D has 3 attackers: B_2, C, and E. However, we are able to find more than 3 attackers for argument Z: one argument with conclusion C deriving from $C \leftarrow \texttt{not}\ A$, one argument with conclusion E deriving from $E \leftarrow \texttt{not}\ E$, and at least two arguments with conclusion B_2 (namely, X and Y). Clearly, there is no argument in $\mathcal{B}$ with more than 3 attackers. This mismatch shows why this approach fails to preserve the structure of the original *BAF*.

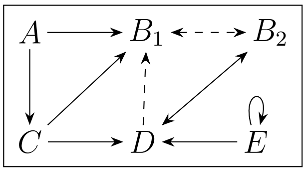
Inspired by the problems aforementioned, we define corresponding *BAF*s in a more suitable way for establishing syntactic correspondences. Most of the added complexity serves to avoid redundancies, which in turn allows preserving structure.

Definition 37 (Corresponding *NLP*)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. For any $A, B \in \mathcal{A}$ with $B \in \mathfrak{Sup}(A)$, define the rule $r_{A,B}$ such that $\mathit{head}(r_{A,B}) = \mathfrak{Sup}(A)$, $\mathit{body}^+(r_{A,B}) = \emptyset$, and $\mathit{body}^-(r_{A,B}) = \{\texttt{not}\ \mathfrak{Sup}(X) \mid X \in \mathit{Att}(B)\}$. The corresponding *NLP* $P_{\mathcal{B}}$ of $\mathcal{B}$ is the program with the set of rules

$$P_{\mathcal{B}} = \{r_{A,B} \mid A \in \mathcal{A}, B \in \mathfrak{Sup}(A)\}.$$

Each argument is encoded in the corresponding *NLP* by its set of supporters and the Herbrand Base of $P_{\mathcal{B}}$ is $\mathit{HB}_{P_{\mathcal{B}}} = \{\mathfrak{Sup}(A) \mid A \in \mathcal{A}\}$. Intuitively, with respect to any partial stable model of $P_{\mathcal{B}}$, we know $\mathfrak{Sup}(A)$ is true iff there exists $B \in \mathfrak{Sup}(A)$ such that $\mathfrak{Sup}(X)$ is false for every $X \in \mathit{Att}(B)$. This property follows directly from Definition 37, that was conceived to reflect the concept of acceptability in the β-complete semantics, where an argument $A \in \mathcal{A}$ is accepted iff there exists $B \in \mathfrak{Sup}(A)$ such that X is rejected for every $X \in \mathit{Att}(B)$.

Example 16. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be the *BAF* of Figure 23a, and $P_{\mathcal{B}}$ be its corresponding *NLP* shown in Figure 23b.



(a) $\mathcal{B}$

$$
\begin{aligned}
&\{A\} \\
&\{B_1, B_2, D\} \leftarrow \texttt{not}\ \{A\},\ \texttt{not}\ \{C\} \\
&\{B_1, B_2, D\} \leftarrow \texttt{not}\ \{D\} \\
&\{B_1, B_2, D\} \leftarrow \texttt{not}\ \{B_1, B_2, D\},\ \texttt{not}\ \{C\},\ \texttt{not}\ \{E\} \\
&\{C\} \leftarrow \texttt{not}\ \{A\} \\
&\{D\} \leftarrow \texttt{not}\ \{B_1, B_2, D\},\ \texttt{not}\ \{C\},\ \texttt{not}\ \{E\} \\
&\{E\} \leftarrow \texttt{not}\ \{E\}
\end{aligned}
$$

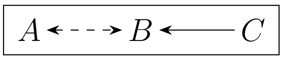
(b) $P_{\mathcal{B}}$

Figure 23 – (a) *BAF* $\mathcal{B}$ from Example 16 and (b) its corresponding *NLP* $P_{\mathcal{B}}$.

Note that distinct arguments A and A' are associated to the same atom if $\mathfrak{Sup}(A) = \mathfrak{Sup}(A')$. For instance, the corresponding *NLP* of Example 23b encodes both arguments B_1 and B_2 as the atom $\mathfrak{Sup}(B_1) = \mathfrak{Sup}(B_2) = \{B_1, B_2, D\}$. From the supporters of B_1, we obtain the rules $r_{B_1,B_1} = \{B_1, B_2, D\} \leftarrow \texttt{not}\ \{A\}, \texttt{not}\ \{C\}$; $r_{B_1,B_2} = \{B_1, B_2, D\} \leftarrow \texttt{not}\ \{D\}$; and $r_{B_1,D} = \{B_1, B_2, D\} \leftarrow \texttt{not}\ \{B_1, B_2, D\}, \texttt{not}\ \{C\}, \texttt{not}\ \{E\}$. The intuition is that B_1 is accepted iff either (i) A and C are rejected or (ii) D is rejected or (iii) B_1, B_2, C, D, and E are rejected. Condition (iii) seems contradictory, but it is similar to what occurs when an argument attacks itself (e.g., argument E is accepted iff E is rejected).

As done in the previous section, we want to establish a bijection between labelings and interpretations, such that β-complete labelings correspond to partial stable models. However, there can be more labelings than interpretations.

Example 17. Let $\mathcal{B}' = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be the *BAF* depicted below.



The corresponding *NLP* $P_{\mathcal{B}'}$ is

$$
\begin{aligned}
&\{A, B\} \leftarrow \texttt{not}\ \{C\} \\
&\{A, B\} \\
&\{C\}.
\end{aligned}
$$

There are $3^{|\mathcal{A}|} = 27$ labelings of $\mathcal{B}'$ and $3^{|HB_{P_{\mathcal{B}'}}|} = 9$ interpretations of $P_{\mathcal{B}'}$. Hence, there is no bijection between labelings of $\mathcal{B}'$ and interpretations of $P_{\mathcal{B}'}$. Notice that a labeling may assign different labels to A and B, whereas an interpretation assigns only one truth value to the atom $\{A, B\}$.

Although there is no bijection between labelings and interpretations for the example above, there is always a bijection if we restrict our translations to labelings respecting $\mathfrak{Sup}$, defined next.

Definition 38. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. A labeling $\mathcal{L}$ of $\mathcal{B}$ respects $\mathfrak{Sup}$ iff $\mathcal{L}(A) = \mathcal{L}(B)$ for every $A, B \in \mathcal{A}$ with $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$.

In the direction from *NLP*s to *BAF*s (Definition 35), arguments with the same conclusion mutually support each other, which in turn means they have the same set of supporters. Then we represent arguments by their set of supporters in the direction from *BAF*s to *NLP*s (Definition 37), as arguments with identical set of supporters share the same label in any β-complete labeling:

Proposition 43. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$, then $\mathcal{L}$ respects $\mathfrak{Sup}$.

For the *BAF* $\mathcal{B}'$ of Example 17, there are 9 labelings respecting $\mathfrak{Sup}$, since a labeling that respects $\mathfrak{Sup}$ assigns the same label to A and B. The representation of possibly many arguments by the same atom is motivated by Proposition 43.

By restricting labelings to those that respect $\mathfrak{Sup}$, the relationship between labelings and interpretations is straightforward:

Definition 39 ($\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}$ and $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}$ functions)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $P_\mathcal{B}$ be its corresponding *NLP*. Denote the set of all labelings of $\mathcal{B}$ respecting $\mathfrak{Sup}$ as $\mathcal{L}ab^*$ and the set of all interpretations of $P_\mathcal{B}$ as $\mathcal{I}nt$. We introduce the functions below:

a) $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}} : \mathcal{L}ab^* \to \mathcal{I}nt$, in which for every $\mathcal{L} \in \mathcal{L}ab^*$,

$$\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{t} \quad \text{if } \mathcal{L}(A) = \texttt{in}$$

$$\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{f} \quad \text{if } \mathcal{L}(A) = \texttt{out}$$

$$\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{u} \quad \text{if } \mathcal{L}(A) = \texttt{undec};$$

b) $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}} : \mathcal{I}nt \to \mathcal{L}ab^*$, in which for every $\mathcal{M} \in \mathcal{I}nt$,

$$\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{M})(A) = \texttt{in} \quad \text{if } \mathcal{M}(\mathfrak{Sup}(A)) = \mathbf{t}$$

$$\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{M})(A) = \texttt{out} \quad \text{if } \mathcal{M}(\mathfrak{Sup}(A)) = \mathbf{f}$$

$$\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{M})(A) = \texttt{undec} \quad \text{if } \mathcal{M}(\mathfrak{Sup}(A)) = \mathbf{u}.$$

The functions are well-defined only for labelings respecting $\mathfrak{Sup}$. Semantics-wise, this is a minor restriction, as every labeling under the β-complete, β-grounded, β-preferred, β-stable, or β-semi-stable semantics respects $\mathfrak{Sup}$.

Unlike $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}$ and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}$, the functions $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}$ and $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}$ are the inverse of each other in the general case:

Theorem 44. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $P_\mathcal{B}$ be its corresponding *NLP*. The following holds:

a) for any labeling $\mathcal{L}$ of $\mathcal{B}$ respecting $\mathfrak{Sup}$, it holds $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})) = \mathcal{L}$;
b) for any interpretation $\mathcal{I}$ of $P_\mathcal{B}$, it holds $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{I})) = \mathcal{I}$.

Note that the assumption that $\mathcal{L}$ must respect $\mathfrak{Sup}$ is already given by $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}$'s domain. The next lemma is essential for the correspondence between β-complete labelings of $\mathcal{B}$ and partial stable models of $P_\mathcal{B}$.

Lemma 45. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $P_\mathcal{B}$ be its corresponding *NLP*. Let $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$ and $\mathcal{M}$ be an interpretation of $P_\mathcal{B}$. If $\mathcal{L} = \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{M})$ or $\mathcal{M} = \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$, then for any $A \in \mathcal{A}$, the following holds:

a) there exists $B \in \mathfrak{Sup}(A)$ such that $Att(B) \subseteq \texttt{out}(\mathcal{L})$ iff there exists $r \in P_\mathcal{B}$ with $head(r) = \mathfrak{Sup}(A)$ such that $\{\mathfrak{Sup}(X) \mid \texttt{not}\ \mathfrak{Sup}(X) \in body^-(r)\} \subseteq \mathbf{f}(\mathcal{M})$;
b) for every $B \in \mathfrak{Sup}(A)$ it holds $Att(B) \cap \texttt{in}(\mathcal{L}) \neq \emptyset$ iff for every rule $r \in P_\mathcal{B}$ with $head(r) = \mathfrak{Sup}(A)$ it holds $\{\mathfrak{Sup}(X) \mid \texttt{not}\ \mathfrak{Sup}(X) \in body^-(r)\} \cap \mathbf{t}(\mathcal{M}) \neq \emptyset$.

From Lemma 45, we obtain a similar result to Theorem 40:

Theorem 46. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*, $P_\mathcal{B}$ be its corresponding *NLP*, $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$ and $\mathcal{M}$ be an interpretation of $P_\mathcal{B}$. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a partial stable model of $P_\mathcal{B}$.
b) $\mathcal{M}$ is a partial stable model of $P_\mathcal{B}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}$.

From Theorem 46, we can ensure the equivalence between the semantics for *BAF*s and their counterpart for *NLP*s:

Theorem 47. Let $\mathcal{B}$ be a *BAF* and $P_\mathcal{B}$ be its corresponding *NLP*. For any labeling $\mathcal{L}$ of $\mathcal{B}$ respecting $\mathfrak{Sup}$, the following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a well-founded model of $P_\mathcal{B}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is a regular model of $P_{\mathcal{B}}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is a stable model of $P_{\mathcal{B}}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is an L-stable model of $P_{\mathcal{B}}$.

The following result is a direct consequence of Theorems 44 and 47:

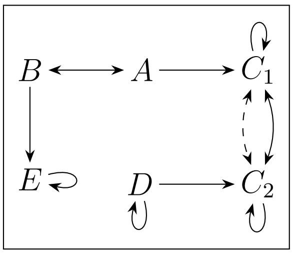
Corollary 48. Let $\mathcal{B}$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. The following holds:

a) $\mathcal{M}$ is a well-founded model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-grounded labeling of $\mathcal{B}$;

b) $\mathcal{M}$ is a regular model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-preferred labeling of $\mathcal{B}$;

c) $\mathcal{M}$ is a stable model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-stable labeling of $\mathcal{B}$;

d) $\mathcal{M}$ is a semi-stable model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-semi-stable labeling of $\mathcal{B}$.



(a) *BAF* from Example 14

$\{C_1, C_2\} \leftarrow \texttt{not}\ \{C_1, C_2\},\ \texttt{not}\ \{D\}$
$\{C_1, C_2\} \leftarrow \texttt{not}\ \{C_1, C_2\},\ \texttt{not}\ \{A\}$
$\{E\} \leftarrow \texttt{not}\ \{E\},\ \texttt{not}\ \{B\}$
$\{D\} \leftarrow \texttt{not}\ \{D\}$
$\{B\} \leftarrow \texttt{not}\ \{A\}$
$\{A\} \leftarrow \texttt{not}\ \{B\}$

(b) Its corresponding *NLP*

Figure 24 – *BAF* from Example 14 and its corresponding *NLP*.

β-Complete labelings	Partial Stable Models
$\mathcal{L}_1 = (\emptyset, \emptyset, \{A, B, C_1, C_2, D, E\})$	$\mathcal{M}_1 = (\emptyset, \emptyset, \{\{A\}, \{B\}, \{C_1, C_2\}, \{D\}, \{E\}\})$
$\mathcal{L}_2 = (\{A\}, \{B\}, \{C_1, C_2, D, E\})$	$\mathcal{M}_2 = (\{\{A\}\}, \{\{B\}\}, \{\{C_1, C_2\}, \{D\}, \{E\}\})$
$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$	$\mathcal{M}_3 = (\{\{B\}\}, \{\{A\}, \{E\}\}, \{\{C_1, C_2\}, \{D\}\})$
β-Grounded labeling	Well-Founded Model
$\mathcal{L}_1 = (\emptyset, \emptyset, \{A, B, C_1, C_2, D, E\})$	$\mathcal{M}_1 = (\emptyset, \emptyset, \{\{A\}, \{B\}, \{C_1, C_2\}, \{D\}, \{E\}\})$
β-Preferred labelings	Regular Models
$\mathcal{L}_2 = (\{A\}, \{B\}, \{C_1, C_2, D, E\})$	$\mathcal{M}_2 = (\{\{A\}\}, \{\{B\}\}, \{\{C_1, C_2\}, \{D\}, \{E\}\})$
$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$	$\mathcal{M}_3 = (\{\{B\}\}, \{\{A\}, \{E\}\}, \{\{C_1, C_2\}, \{D\}\})$
β-Stable labelings	Stable Models
None	None
β-Semi-stable labelings	L-Stable Models
$\mathcal{L}_3 = (\{B\}, \{A, E\}, \{C_1, C_2, D\})$	$\mathcal{M}_3 = (\{\{B\}\}, \{\{A\}, \{E\}\}, \{\{C_1, C_2\}, \{D\}\})$

Table 5 – Semantics for the *BAF* and corresponding *NLP* from Figure 24.

From the *BAF* of Example 14 and its corresponding *NLP*, both depicted in Figure 24, we obtain the expected equivalence results related to their semantics (see Table 5). Observe that from the *NLP* P of Example 14 (Figure 21a), we constructed the corresponding *BAF* $\mathcal{B}_P$

(Figures 21b and 24a) and then its corresponding *NLP* $P_{\mathcal{B}_P}$ (Figure 24b). It is no coincidence that P and $P_{\mathcal{B}_P}$ are isomorphic, i.e., each can be obtained by renaming the atoms from the other. In the next section, we will find a class of *BAF*s and *NLP*s such that the translation from an *NLP* to a *BAF* (Definition 35) behaves as the inverse (up to isomorphism) of the translation from a *BAF* to an *NLP* (Definition 37).

4.4 On the relation between *BAF*s and *NLP*s

In preceding sections, we have shown how to translate *BAF*s to *NLP*s, and vice versa, while preserving their corresponding semantics. Now we check under which conditions these translations are the inverses (up to isomorphism) of each other, showing that *BAF*s and *NLP*s under these conditions are connected at a syntactic level. We say *BAF*s $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}' = (\mathcal{A}', \mathit{Att}', \mathit{Sup}')$ are isomorphic if there exists a bijection $f : \mathcal{A} \to \mathcal{A}'$ such that all conditions below hold:

a) $\{f(A) \mid A \in \mathcal{A}\} = \mathcal{A}'$;

b) $\{(f(A), f(B)) \mid (A, B) \in \mathit{Att}\} = \mathit{Att}'$;

c) $\{(f(A), f(B)) \mid (A, B) \in \mathit{Sup}\} = \mathit{Sup}'$.

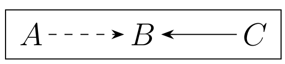
Similarly, *NLP*s P and P' are isomorphic when there is a bijection $f : \mathit{HB}_P \to \mathit{HB}_{P'}$ such that $\{f(c) \leftarrow f(a_1), \ldots, f(a_m), \texttt{not}\ f(b_1), \ldots, \texttt{not}\ f(b_n) \mid c \leftarrow a_1, \ldots, a_m, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n \in P\} = P'$.

From a *BAF* $\mathcal{B}$, we obtain its corresponding *NLP* $P_{\mathcal{B}}$ via Definition 37; from $P_{\mathcal{B}}$, we obtain its corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$ via Definition 35. By following the other direction, from an *NLP* P, we obtain its corresponding *BAF* $\mathcal{B}_P$, and from $\mathcal{B}_P$, its corresponding *NLP* $P_{\mathcal{B}_P}$. In this section, we investigate for which class of *BAF*s we have $\mathcal{B}$ is isomorphic to $\mathcal{B}_{P_{\mathcal{B}}}$, and for which class of *NLP*s the programs P and $P_{\mathcal{B}_P}$ are isomorphic.

4.4.1 On the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$

Initially, we show that, in general, $\mathcal{B}$ is not isomorphic to $\mathcal{B}_{P_{\mathcal{B}}}$. This result is expected, as by Definition 35, every corresponding *BAF* of an *NLP* has a symmetric support relation.

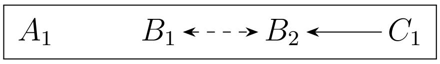
Example 18. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be the following *BAF*:



Its corresponding *NLP* $P_{\mathcal{B}}$ is

$$\{A\} \qquad\qquad\qquad\qquad \{C\}$$
$$\{A, B\} \qquad\qquad \{A, B\} \leftarrow \texttt{not}\ \{C\}.$$

Its corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$ is shown below, where arguments A_1, B_1, B_2, and C_1 are derived respectively from the rules $\{A\}$, $\{A, B\}$, $\{A, B\} \leftarrow \texttt{not}\ \{C\}$, and $\{C\}$.



Notice that the support relation of $\mathcal{B}_{P_{\mathcal{B}}}$ is symmetric, whereas the support relation of $\mathcal{B}$ is not. Clearly, they are not isomorphic. However, symmetry is not the only decisive factor for determining whether $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are isomorphic, i.e., they can be non-isomorphic even if $\mathcal{B}$ has a symmetric support relation. This occurs because the support relation of a *BAF* $\mathcal{B}_P$ that corresponds to some *NLP* P is very specific: distinct arguments support each other in $\mathcal{B}_P$ exactly when they derive from rules in P sharing the same conclusion. An important property of every *BAF* that corresponds to some *NLP* is defined next:

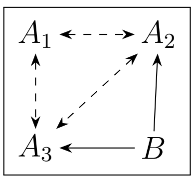
Definition 40 (*BAF* of support cliques, $\mathfrak{S}$-*BAF*)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. We say $\mathcal{B}$ is a *BAF* of support cliques ($\mathfrak{S}$-*BAF*) iff *Sup* is irreflexive, symmetric, and for any $(A, B), (B, C) \in$ *Sup* with $A \neq C$, it holds $(A, C) \in$ *Sup*.

The name *support cliques* comes from the fact that we could have equivalently defined it as a *BAF* for which *Sup* is irreflexive, symmetric, and $\mathcal{A}$ can be partitioned into maximal cliques of the graph $(\mathcal{A}, \mathit{Sup})$, where a maximal clique is a subset $S \subseteq \mathcal{A}$ closed for *Sup* and such that any distinct $A, B \in S$ are adjacent (by *Sup*).

Lemma 49. For any *NLP* P, the corresponding *BAF* $\mathcal{B}_P$ is a $\mathfrak{S}$-*BAF*.

There are *BAF*s of support cliques that can only correspond to "redundant" *NLP*s (as we shall define later).

Example 19. Consider the following $\mathfrak{S}$-*BAF*:



Suppose there exists an *NLP* P such that $\textit{body}^+(r) = \emptyset$ for every rule $r \in P$ and whose corresponding *BAF* is the one of this example. As $(A_2, A_3) \in \textit{Sup}$, we obtain $A_2 \neq A_3$ and $\texttt{Conc}(A_2) = \texttt{Conc}(A_3)$. Let $r_2, r_3 \in P$ be the rules that derived A_2 and A_3, respectively. As $\textit{Att}(A_2) = \textit{Att}(A_3)$, we obtain $\textit{body}(r_2) = \textit{body}(r_3)$. As $\texttt{Conc}(A_2) = \texttt{Conc}(A_3)$, it follows $\textit{head}(r_2) = \textit{head}(r_3)$. Therefore, $r_2 = r_3$, which means that $A_2 = A_3$, as only one argument can be derived from a rule r with $\textit{body}^+(r) = \emptyset$. As $A_2 \neq A_3$, we found an absurd. Hence, we have shown that any *NLP* that corresponds to this *BAF* has some positive body atom (i.e., $\textit{body}^+(r) \neq \emptyset$ for some rule r), which will be analyzed in Subsection 4.4.2 as a kind of redundancy.

That occurred because arguments A_2 and A_3 share the same set of supporters and attackers. Recall that if arguments A and B satisfy $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, then A is labeled the same as B for any β-complete labeling. Therefore, attackers of an argument A represent a possible condition to accept any argument with the same supporters as those of A. More specifically, the acceptance condition is that every attacker of A is rejected. Arguments with identical sets of supporters and attackers are, in that sense, redundant:

Definition 41 (Redundancy-free *BAF*, *RFBAF*)**.** Let $\mathcal{B} = (\mathcal{A}, \textit{Att}, \textit{Sup})$ be a *BAF*. Arguments $A, B \in \mathcal{A}$ are redundant iff $A \neq B$ and $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and $\textit{Att}(A) = \textit{Att}(B)$. When a *BAF* $\mathcal{B}$ contains redundant arguments, we say $\mathcal{B}$ is redundant. Otherwise, $\mathcal{B}$ is a redundancy-free *BAF* (*RFBAF*).

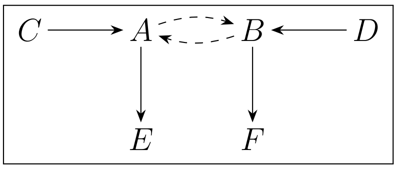
When a *RFBAF* is also a $\mathfrak{S}$-*BAF*, we call it a redundancy-free $\mathfrak{S}$-*BAF*, denoted $\mathfrak{S}$-*RFBAF*. As this redundancy arises from the support relation, no *AAF* is redundant.

Proposition 50. Let $\mathcal{B} = (\mathcal{A}, \textit{Att}, \textit{Sup})$ be a *BAF*. If $\textit{Sup} = \emptyset$, then $\mathcal{B}$ is a *RFBAF*.

Moreover, this concept of redundancy aligns with the proposed translation from *BAF*s to *NLP*s, as redundant arguments derive the same set of rules. For instance, arguments A_2 and A_3 in the *BAF* of Example 19 have the same set of supporters $\{A_1, A_2, A_3\}$ and set of attackers $\{B\}$, and derive the same rule $\{A_1, A_2, A_3\} \leftarrow \texttt{not}\ \{B\}$. Notice the loss of information: two arguments derive one unique rule, and from this one rule we cannot recover the original two arguments.

Next, we discuss one more criterion necessary for the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$.

Example 20. Let $\mathcal{B} = (\mathcal{A}, \textit{Att}, \textit{Sup})$ be the following *BAF*:



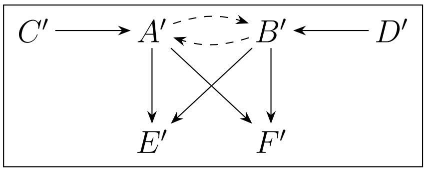
The corresponding *NLP* $P_{\mathcal{B}}$ is

$$\{A, B\} \leftarrow \texttt{not}\ \{C\} \qquad \{D\} \leftarrow$$

$$\{A, B\} \leftarrow \texttt{not}\ \{D\} \qquad \{E\} \leftarrow \texttt{not}\ \{A, B\}$$

$$\{C\} \leftarrow \qquad \{F\} \leftarrow \texttt{not}\ \{A, B\}.$$

The corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$ is shown below:



We found 6 arguments in the corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$, where A' is $\{A, B\} \leftarrow \texttt{not}\ \{C\}$; B' is $\{A, B\} \leftarrow \texttt{not}\ \{D\}$; C' is $\{C\} \leftarrow$; D' is $\{D\} \leftarrow$; E' is $\{E\} \leftarrow \texttt{not}\ \{A, B\}$; and F' is $\{F\} \leftarrow \texttt{not}\ \{A, B\}$. Notably, both arguments A' and B' attack the same arguments: namely, E' and F'. Hence, there are 6 attacks in $\mathcal{B}_{P_{\mathcal{B}}}$, whereas $\mathcal{B}$ only has 4 attacks. Clearly, $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are not isomorphic.

This happened because, in $\mathcal{B}$, we have $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, but A and B do not attack the same arguments. This motivates the definition of a *BAF* of support-guided attacks:

Definition 42 (*BAF* of support-guided attacks)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. If A and B attack the same arguments for any $A, B \in \mathcal{A}$ with $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, then we say $\mathcal{B}$ is of support-guided attacks.

By Definition 34, any corresponding *BAF* is of support-guided attacks.

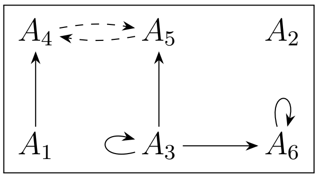
Proposition 51. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. It holds $\mathcal{B}_P$ is of support-guided attacks.

This is useful, because it means that the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ can only hold if $\mathcal{B}$ is a *BAF* of support-guided attacks. Now, we prove that the isomorphism holds precisely for the class of $\mathfrak{S}$-*RFBAF*s of support-guided attacks. For convenience, we will give this class a name.

Definition 43 ($\mathfrak{S}^+$-*RFBAF*)**.** Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. We say $\mathcal{B}$ is a $\mathfrak{S}^+$-*RFBAF* if $\mathcal{B}$ is a $\mathfrak{S}$-*RFBAF* of support-guided attacks.

Now we prove that the isomorphism holds precisely for the class of $\mathfrak{S}^+$-*RFBAF*s:

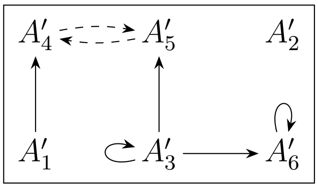
Theorem 52. Let $\mathcal{B}$ be a *BAF*, $P_{\mathcal{B}}$ be its corresponding *NLP*, and $\mathcal{B}_{P_{\mathcal{B}}}$ be the corresponding *BAF* of $P_{\mathcal{B}}$. It holds $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are isomorphic iff $\mathcal{B}$ is a $\mathfrak{S}^+$-*RFBAF*.



(a) $\mathcal{B}$

$$\begin{array}{l} \{A_1\} \\ \{A_2\} \\ \{A_4, A_5\} \leftarrow \texttt{not}\ \{A_1\} \\ \{A_4, A_5\} \leftarrow \texttt{not}\ \{A_3\} \\ \{A_3\} \leftarrow \texttt{not}\ \{A_3\} \\ \{A_6\} \leftarrow \texttt{not}\ \{A_3\}, \texttt{not}\ \{A_6\} \end{array}$$

(b) $P_{\mathcal{B}}$



(c) $\mathcal{B}_{P_{\mathcal{B}}}$

Figure 25 – *BAF* $\mathcal{B}$, its corresponding *NLP* $P_{\mathcal{B}}$, and the corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$ of $P_{\mathcal{B}}$. Arguments $A'_1, A'_2, A'_3, A'_4, A'_5, A'_6$ are respectively derived from the rules $\{A_1\}$, $\{A_2\}$, $\{A_3\} \leftarrow \texttt{not}\ \{A_3\}$, $\{A_4, A_5\} \leftarrow \texttt{not}\ \{A_1\}$, $\{A_4, A_5\} \leftarrow \texttt{not}\ \{A_3\}$, and $\{A_6\} \leftarrow \texttt{not}\ \{A_3\}, \texttt{not}\ \{A_6\}$.

Example 21. Let $\mathcal{B}$ be the *BAF* of Example 13, depicted again in Figure 25 for convenience, next to its corresponding *NLP* $P_{\mathcal{B}}$ and the corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$ of $P_{\mathcal{B}}$. As expected, $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are isomorphic.

4.4.2 On the isomorphism between P and $P_{\mathcal{B}_P}$

Now we investigate under which conditions P and $P_{\mathcal{B}_P}$ are isomorphic. Example 22 shows that some *NLP*s correspond to the same *BAF* (up to isomorphism).

Example 22. Let P be the *NLP* with rules $\{r_1 = a, r_2 = b\}$, P' be the *NLP* with rules $\{r'_1 = a, r'_2 = b \leftarrow \texttt{not}\ c\}$, and P'' be the *NLP* with rules $\{r''_1 = a, r''_2 = b, r''_3 = d \leftarrow c\}$.

Notice that these *NLP*s are almost the same: P is almost equal to P', except for the occurrence of $\texttt{not}\ c$ in the body of r'_2; and P is almost equal to P'', except for the rule $r''_3 \in P''$.

In the corresponding *BAF* of an *NLP* P, there is no argument with conclusion c such that $c \in \mathit{HB}_P - \{\mathit{head}(r) \mid r \in P\}$. Then, according to Definition 35, the literal $\mathtt{not}\ c$ is ignored when $\mathtt{not}\ c \in \mathit{body}^-(r)$, and the whole rule r is ignored when $c \in \mathit{body}^+(r)$. It follows that some rules are effectively the same, as they derive the same arguments in the corresponding *BAF*. For instance, $\mathtt{not}\ c$ is ignored in $r_2' = b \leftarrow \mathtt{not}\ c$, making r_2' effectively the same as $r_2 = b$; and $r_3'' = d \leftarrow c$ is ignored in P'', making P'' effectively the same as P.

As such, the corresponding *BAF*s of each of these *NLP*s are all isomorphic to the *BAF* $(\{A_0, A_1\}, \emptyset, \emptyset)$, with only two arguments and no attacks/supports. Clearly, the corresponding *NLP* of $(\{A_0, A_1\}, \emptyset, \emptyset)$ cannot be isomorphic to all of P, P', and P''. In fact, it is isomorphic to P, which has no ignored atoms or rules, as every atom appears in the head of some rule.

The hint given by the example above is that the isomorphism only holds for *NLP*s such that every atom appears in the head of some rule.

Lemma 53. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF*, and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. If $\mathit{HB}_P \neq \{\mathit{head}(r) \mid r \in P\}$, then P and $P_{\mathcal{B}_P}$ are not isomorphic.

There is another scenario in which the isomorphism does not hold. Note that, by Definition 37, the corresponding *NLP* P' of some *BAF* $\mathcal{B}$ has only rules r with empty $\mathit{body}^+(r)$. In particular, if P has a rule r with nonempty $\mathit{body}^+(r)$, $P_{\mathcal{B}_P}$ cannot be isomorphic to P. This is the same kind of redundancy as the one mentioned in Subsection 4.4.1.

Lemma 54. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF* and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. If there is some $r \in P$ such that $\mathit{body}^+(r) \neq \emptyset$, then P and $P_{\mathcal{B}_P}$ are not isomorphic.

From the two conditions discussed above, we define the class of Reduced Atomic Logic Programs (*RALP*s):

Definition 44 (Reduced Atomic Logic Program, *RALP*)**.** We say an *NLP* P is a Reduced Atomic Logic Program (*RALP*) if for any rule $r \in P$, it holds $\mathit{body}^+(r) = \emptyset$, and $\mathit{HB}_P = \{\mathit{head}(r) \mid r \in P\}$.

We find that *RALP*s are precisely the *NLP*s for which the isomorphism between P and $P_{\mathcal{B}_P}$ holds.

Theorem 55. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF* and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. It holds P and $P_{\mathcal{B}_P}$ are isomorphic iff P is an *RALP*.

Example 23. Recall the *NLP* P and its corresponding *BAF* $\mathcal{B}_P$ of Example 14. They are depicted again in Figure 26 for convenience, together with the corresponding *NLP* of $\mathcal{B}_P$. Notice the isomorphism between P and $P_{\mathcal{B}_P}$.

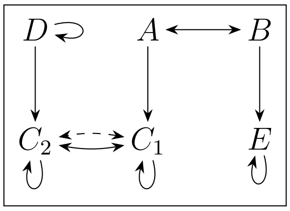
The corresponding *NLP* $P_{\mathcal{B}_P}$ encodes arguments that have identical sets of supporters (e.g., C_1, C_2) as only one atom (e.g., $\{C_1, C_2\}$). This is essential for the isomorphism between P and $P_{\mathcal{B}_P}$, as $\mathcal{B}_P$ encodes each atom of P as possibly many arguments sharing the same set of supporters (e.g., from atom c we obtain arguments C_1 and C_2). The isomorphism comes from the collapse of C_1 and C_2 into their set of supporters $\{C_1, C_2\}$.

$$
\begin{array}{l}
c \leftarrow \texttt{not}\ c, \texttt{not}\ d \\
c \leftarrow \texttt{not}\ c, \texttt{not}\ a \\
e \leftarrow \texttt{not}\ e, \texttt{not}\ b \\
d \leftarrow \texttt{not}\ d \\
a \leftarrow \texttt{not}\ b \\
b \leftarrow \texttt{not}\ a
\end{array}
$$

(a) P

$$
\begin{array}{c}
\{C_1, C_2\} \leftarrow \texttt{not}\ \{C_1, C_2\}, \texttt{not}\ \{D\} \\
\{C_1, C_2\} \leftarrow \texttt{not}\ \{C_1, C_2\}, \texttt{not}\ \{A\} \\
\{E\} \leftarrow \texttt{not}\ \{E\}, \texttt{not}\ \{B\} \\
\{D\} \leftarrow \texttt{not}\ \{D\} \\
\{A\} \leftarrow \texttt{not}\ \{B\} \\
\{B\} \leftarrow \texttt{not}\ \{A\}
\end{array}
$$

(b) $P_{\mathcal{B}_P}$



(c) $\mathcal{B}_P$

Figure 26 – *NLP* P, its corresponding *BAF* $\mathcal{B}_P$, and the corresponding *NLP* $P_{\mathcal{B}_P}$ of $\mathcal{B}_P$. Arguments A, B, C_1, C_2, D, E are respectively derived from the rules $(a \leftarrow \texttt{not}\ b)$, $(b \leftarrow \texttt{not}\ a)$, $(c \leftarrow \texttt{not}\ c, \texttt{not}\ a)$, $(c \leftarrow \texttt{not}\ c, \texttt{not}\ d)$, $(d \leftarrow \texttt{not}\ d)$ and $(e \leftarrow \texttt{not}\ e, \texttt{not}\ b)$.

4.5 On the relation between *NLP*s and *RALP*s

In Section 4.4, we established that Reduced Atomic Logic Programs (*RALP*s) and redundancy-free *BAF*s of support cliques and of support-guided attacks ($\mathfrak{S}^+$-*RFBAF*s) are connected at a syntactic level. Here, we investigate the connection between *NLP*s and *RALP*s, highlighting the link between the corresponding *BAF*s of *NLP*s. Initially, we show that *RALP*s are as expressive as *NLP*s when considering the semantics for *NLP*s exploited in our work. Our idea is to transform any *NLP* P into a unique *RALP* P^* by resorting to the transformations employed in the characterization of the *RALP* $P_{\mathcal{B}_P}$ (see Sections 4.2 and 4.3). These transformations have already been shown to preserve the semantics under consideration. Additionally, by analyzing corresponding *BAF*s, we find many links between *NLP*s and *RALP*s:

a) for any *NLP* P such that $\mathcal{B}_P$ is an *RFBAF*, there is an *RALP* P^* such that $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$ are isomorphic (Theorem 59);
b) *RALP*s P and P' are isomorphic iff $\mathcal{B}_P$ and $\mathcal{B}_{P'}$ are isomorphic (Theorem 60);
c) for any *NLP*s P and P' such that their corresponding *BAF*s are *RFBAF*s, it holds that the corresponding *RALP*s of P and P' are isomorphic iff $\mathcal{B}_P$ and $\mathcal{B}_{P'}$ are isomorphic (Theorem 61).

The assumption that the corresponding *BAF* is an *RFBAF* in the results above is a consequence of Definition 33 allowing the construction of redundant arguments. In the end of this section, we also explain how corresponding *BAF*s could have been alternatively defined, by first transforming an *NLP* into an *RALP* before applying Definition 33 to the resulting *RALP*. As every corresponding *BAF* of an *RALP* is an *RFBAF*, this assumption would no longer be required for Theorems 59 and 61.

We proceed by defining formally expressiveness in terms of signatures of the semantics (Dunne *et al.*, 2015):

Definition 45 (Expressiveness, Dunne *et al.* 2015)**.** Let $\mathcal{P}$ be a class of *NLP*s. The signature $\Sigma^{\mathcal{P}}_{PSM}$ of the partial stable models associated with $\mathcal{P}$ is defined as

$$\Sigma^{\mathcal{P}}_{PSM} = \{\sigma(P) \mid P \in \mathcal{P}\},$$

where $\sigma(P) = \{\mathcal{I} \mid \mathcal{I} \textit{ is a partial stable model of } P\}$ is the set of all partial stable models of P.

Given two classes $\mathcal{P}_1$ and $\mathcal{P}_2$ of *NLP*s, we say that $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same expressiveness for the partial stable models semantics if $\Sigma^{\mathcal{P}_1}_{PSM} = \Sigma^{\mathcal{P}_2}_{PSM}$. In other words, $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same expressiveness if both conditions below hold:

a) for every $P_1 \in \mathcal{P}_1$, there exists $P_2 \in \mathcal{P}_2$ such that P_1 and P_2 have the same set of partial stable models;
b) for every $P_2 \in \mathcal{P}_2$, there exists $P_1 \in \mathcal{P}_1$ such that P_1 and P_2 have the same set of partial stable models.

Similarly, we can define when $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same expressiveness for the well-founded, regular, stable, and L-stable semantics.

As the class of *RALP*s is contained in the class of all *NLP*s, to show that these classes have the same expressiveness for these semantics, it suffices to prove that for every *NLP*, there exists an *RALP* with the same set of partial stable models. We define the corresponding *RALP* of an *NLP* P inspired by the characterization of $P_{\mathcal{B}_P}$ (Sections 4.2 and 4.3):

Definition 46 (Corresponding *RALP*)**.** For any *NLP* P, construct the *NLP* P' inductively as follows:

a) if $r \in P$ and $body^+(r) = \emptyset$, then $r \in P'$ and $\texttt{Rules}(r) = \{r\}$;
b) if there is a rule $r_0 = a \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m \in P$ (with $n > 0$) and for each a_i ($1 \leq i \leq n$) there is some rule $r_i \in P'$ such that $head(r_i) = a_i$ and $r_0 \notin \texttt{Rules}(r_i)$, then $r = a \leftarrow \texttt{not}\ b'_1, \ldots, \texttt{not}\ b'_q \in P'$ with $\{\texttt{not}\ b'_1, \ldots, \texttt{not}\ b'_q\} = \bigcup_{i=0}^{n} body^-(r_i)$ and $\texttt{Rules}(r) = \{r_0\} \cup \bigcup_{i=1}^{n} \texttt{Rules}(r_i)$.

The corresponding *RALP* P^* of P is the result of removing from the body of the rules in P' occurrences of $\texttt{not}\ x$ for every $x \in HB_P - \{head(r) \mid r \in P'\}$.

Example 24. Let P be the *NLP* with the corresponding *RALP* P^* depicted in Figure 27. Atom e is not the head of any rule in P^*, as the rule $e \leftarrow c, f \in P$ contains a positive body atom f which is not the head of any rule in P (and thus the same applies for P'). This behavior mirrors the fact that there is no argument in $\mathcal{B}_P$ with conclusion f (and hence no arguments with conclusion e). Moreover, rule $b \leftarrow a, \texttt{not}\ d$ cannot use $a \leftarrow b, \texttt{not}\ a$ as a subrule, otherwise the latter would be used more than once for deriving a rule in P^*. In addition, $\texttt{not}\ e$ does not occur in P^*, as e is not the head of any rule in P'. This mimics what occurs in constructing arguments of $\mathcal{B}_P$.

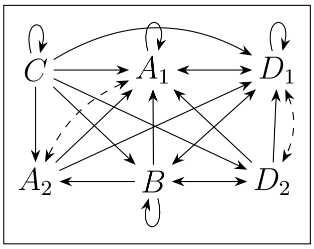
(a) P
$a \leftarrow b, \texttt{not}\ a$
$a \leftarrow c, \texttt{not}\ b$
$b \leftarrow a, \texttt{not}\ d$
$c \leftarrow \texttt{not}\ c$
$d \leftarrow a, \texttt{not}\ e$
$e \leftarrow c, f$

(b) P^*
$a \leftarrow \texttt{not}\ a, \texttt{not}\ b, \texttt{not}\ c, \texttt{not}\ d$
$a \leftarrow \texttt{not}\ b, \texttt{not}\ c$
$b \leftarrow \texttt{not}\ b, \texttt{not}\ c, \texttt{not}\ d$
$c \leftarrow \texttt{not}\ c$
$d \leftarrow \texttt{not}\ a, \texttt{not}\ b, \texttt{not}\ c, \texttt{not}\ d$
$d \leftarrow \texttt{not}\ b, \texttt{not}\ c$

Figure 27 – (a) *NLP* P and (b) its corresponding *RALP* P^*.

The transformations from *NLP* P to *BAF* $\mathcal{B}_P$ (Section 4.2) and from *BAF* $\mathcal{B}_P$ to *NLP* $P_{\mathcal{B}_P}$ (Section 4.3) are the main motivations for defining the corresponding *RALP* P^* of P. Observe how the *RALP*s P^* from Figure 27b and $P_{\mathcal{B}_P}$ from Figure 28b are isomorphic. We show that this is always the case:

Theorem 56. Let P be an *NLP* and P^* be its corresponding *RALP*. It holds P^* and $P_{\mathcal{B}_P}$ are isomorphic.



(a) $\mathcal{B}_P$

$$\{A_1, A_2\} \leftarrow \texttt{not}\ \{A_1, A_2\}, \texttt{not}\ \{B\}, \texttt{not}\ \{C\}, \texttt{not}\ \{D_1, D_2\}$$
$$\{A_1, A_2\} \leftarrow \texttt{not}\ \{B\}, \texttt{not}\ \{C\}$$
$$\{B\} \leftarrow \texttt{not}\ \{B\}, \texttt{not}\ \{C\}, \texttt{not}\ \{D_1, D_2\}$$
$$\{C\} \leftarrow \texttt{not}\ \{C\}$$
$$\{D_1, D_2\} \leftarrow \texttt{not}\ \{A_1, A_2\}, \texttt{not}\ \{B\}, \texttt{not}\ \{C\}, \texttt{not}\ \{D_1, D_2\}$$
$$\{D_1, D_2\} \leftarrow \texttt{not}\ \{B\}, \texttt{not}\ \{C\}$$

(b) $P_{\mathcal{B}_P}$

Figure 28 – (a) corresponding *BAF* $\mathcal{B}_P$ of the *NLP* P from Figure 27a, and (b) its corresponding *NLP* $P_{\mathcal{B}_P}$. Arguments A_1, A_2, B, C, D_1, D_2 are respectively $(a \leftarrow (B), \texttt{not}\ a)$, $(a \leftarrow (C), \texttt{not}\ b)$, $(b \leftarrow (A_2), \texttt{not}\ d)$, $(c \leftarrow \texttt{not}\ c)$, $(d \leftarrow (A_1), \texttt{not}\ e)$, $(d \leftarrow (A_2), \texttt{not}\ e)$.

Consequently, the transformation from *NLP*s to *RALP* preserves the semantics considered in this paper, as P and P^* share the same partial stable models.

Corollary 57. Let P be an *NLP* with corresponding *RALP* P^*. It holds $\mathcal{M}$ is a partial stable, well-founded, regular, stable, and L-stable model of P iff $\mathcal{M}$ is respectively a partial stable, well-founded, regular, stable, and L-stable model of P^*.

Given that each *NLP* can be associated with an *RALP* preserving the semantics above, it follows that *NLP*s and *RALP*s have the same expressiveness for those semantics:

Theorem 58. *NLP*s and *RALP*s have the same expressiveness for partial stable, well-founded, regular, stable, and L-stable semantics.

Another important result we wish to obtain is the isomorphism between $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$, the corresponding *BAF*s of P and P^*, respectively. This does not hold in general, as shown in the next example:

Example 25. Let P be the *NLP*

$$x \leftarrow$$

$$a \leftarrow \texttt{not}\ x$$

$$a \leftarrow a, \texttt{not}\ x$$

with corresponding *RALP* P^*

$$x \leftarrow$$

$$a \leftarrow \texttt{not}\ x.$$

By Definition 33, from P we can construct 3 arguments (namely, $X_1 = x \leftarrow$, $A_1 = a \leftarrow \texttt{not}\ x$, and $A_2 = a \leftarrow (A_1), \texttt{not}\ x$), whereas from P^* we can only construct 2 arguments ($X'_1 = x \leftarrow$ and $A'_1 = a \leftarrow \texttt{not}\ x$).

The problem lies in the presence of two distinct arguments (A_1 and A_2) with identical conclusion a and set of vulnerabilities $\{x\}$. As a consequence, $\mathcal{B}_P$ is not an *RFBAF*. We can avoid this by treating each argument A as the pair $(\texttt{Conc}(A), \texttt{Vul}(A))$. Both A_1 and A_2 would instead be the pair $(a, \{x\})$, coinciding with the only argument constructed from P^*. Using this alternative definition of corresponding *BAF*, the corresponding *BAF* of any *NLP* P would be a $\mathfrak{S}^+$-*RFBAF* (an *RFBAF*, in particular), and we would obtain an equality between the corresponding *BAF*s of P and P^*. In order to avoid introducing another corresponding *BAF* definition, we will instead introduce the assumption that the corresponding *BAF* does not have distinct arguments with identical conclusions and vulnerabilities. Under this assumption, $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$ are isomorphic.

Theorem 59. Let P be an *NLP* and P^* be its corresponding *RALP*. If $\mathcal{B}_P$ is an *RFBAF*, then $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$ are isomorphic.

In addition, as the translations from $\mathfrak{S}^+$-*RFBAF*s to *RALP*s and, conversely, from *RALP*s to $\mathfrak{S}^+$-*RFBAF*s are each other's inverse up to isomorphism (Theorems 52 and 55), we obtain that two non-isomorphic $\mathfrak{S}^+$-*RFBAF*s will always be associated with two non-isomorphic *RALP*s.

Theorem 60. Let P and P' be *RALP*s. It holds P and P' are isomorphic iff $\mathcal{B}_P$ and $\mathcal{B}_{P'}$ are isomorphic.

As the class of *RALP*s is a subset of the class of *NLP*s, many *NLP*s have the same corresponding *BAF*. In particular, *NLP*s with redundancy-free corresponding *BAF*s and with the same corresponding *RALP* have isomorphic corresponding *BAF*s. Next, we prove a more general version of this result:

Theorem 61. Let P_1 and P_2 be *NLP*s with corresponding *RALP*s P_1^* and P_2^*, respectively. If $\mathcal{B}_{P_1}$ and $\mathcal{B}_{P_2}$ are *RFBAF*s, it holds P_1^* and P_2^* are isomorphic iff $\mathcal{B}_{P_1}$ and $\mathcal{B}_{P_2}$ are isomorphic.

The results we have discussed so far also suggest an alternative way to find the corresponding *BAF* of an *NLP* P: instead of resorting directly to Definition 33 to construct arguments, we can first obtain the corresponding *RALP* P^* of P. Then, we apply Definition 33 to this *RALP* to obtain the arguments and Definition 34 for the attack and support relations. Notably, since P^* is an *RALP*, Definition 33 becomes considerably simpler, requiring only its first item to characterize the arguments. Additionally, every argument A of the corresponding *BAF* $\mathcal{B}_P$ in this particular scenario satisfies $|\texttt{Rules}(A)| = 1$ and $\texttt{Sub}(A) = \{A\}$, i.e., argument A can be distinguished from another argument solely by comparing their conclusions and vulnerabilities, as no argument shares the same set of rules or subarguments. This means that the corresponding *BAF*, when employing this strategy, is always an *RFBAF*, allowing this assumption to be removed from Theorems 59 and 61.

Supported by the findings presented in the current section, we can argue that the class of *NLP*s and *RALP*s are as expressive as each other, and are deeply linked to corresponding *BAF*s.

4.6 Conclusions

In this chapter, we extend the work of Caminada; Sá, *et al.* (2015) by considering *BAF*s under the β-semantics proposed in Chapter 3. We leverage the β-semantics for establishing correspondences with the semantics of *NLP*s. We observe that β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable semantics of *BAF*s respectively coincide with the partial stable, well-founded, regular, stable, and L-stable semantics of *NLP*s.

The proposed translations from *NLP*s to *BAF*s (Section 4.2), and vice versa (Section 4.3), show that the inclusion of support in argumentation allows the connection between β-semi-stable and L-stable semantics, whereas Caminada; Sá, *et al.*'s (2015) translation of *AAF*s into *NLP*s does not guarantee a correspondence between semi-stable and L-stable semantics for attack-only argumentation frameworks. Moreover, although an *NLP* can be translated into an *AAF*, recovering the original *NLP* from the corresponding *AAF* is generally not possible in their approach. In contradistinction, our work reveals structural equivalences between some class of *BAF*s and some class of *NLP*s, respectively redundancy-free *BAF*s of support cliques and of support-guided attacks ($\mathfrak{S}^+$-*RFBAF*s) and Reduced Atomic Logic Programs (*RALP*s) (discussed in Section 4.4), which enables the recovery of a *BAF* with the same attack and support relations as the original *BAF*, except for the arguments' names:

- from the corresponding *NLP* $P_{\mathcal{B}}$ of a $\mathfrak{S}^+$-*RFBAF* $\mathcal{B}$, we can obtain the corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$, which is isomorphic to $\mathcal{B}$;
- from the corresponding *BAF* $\mathcal{B}_P$ of an *RALP* P, we can obtain the corresponding *NLP* $P_{\mathcal{B}_P}$, which is isomorphic to P.

Hence, the relationship between *NLP*s and *BAF*s is demonstrably more robust than that between *NLP*s and *AAF*s, extending beyond semantics to encompass structural aspects.

In Section 4.5, we explain how to translate *NLP*s into *RALP*s and prove that both classes have the same expressiveness under the semantics studied in our work. This also suggests an alternative way of finding the corresponding *BAF* of an *NLP* P: instead of constructing arguments directly from P, we can first obtain the corresponding *RALP* P^* of P and then obtain arguments from P^*. Since P^* is an *RALP*, the construction presented in Definition 33 becomes considerably simpler, requiring only its first item to characterize the arguments.

In summary, $\mathfrak{S}^+$-*RFBAF*s and *RALP*s (which are as expressive as *NLP*s) are deeply linked at a syntactic level, each encoding knowledge from its own perspective: argumentation emphasizes the dynamics of arguments and their interactions, whereas *NLP*s focus on deriving atoms from rules. We provide explicit translations between these formalisms and their semantics, allowing for their interchangeable use and enhancing our understanding of their interrelations.

This chapter also allows connecting *BAF*s to other argumentation formalisms: as there is a translation from *BAF*s (β-complete labelings) to *NLP*s (partial stable models), and from *NLP*s (partial stable models) to *ADF*s (Alcântara; Sá; Acosta-Guadarrama, 2019) (complete labelings), one can indirectly obtain a translation from *BAF*s (β-complete labelings) to *ADF*s (complete labelings).

Next, we establish connections between *SETAF*s and *BAF*s under the β-semantics.

5 ON THE EQUIVALENCE BETWEEN BIPOLAR ARGUMENTATION AND *SETAF*

In this chapter, we show the notions of support from the β-semantics and of collective attacks in argumentation are equivalent, by providing direct translations from BAFs and SETAFs and vice versa in a one-to-one correspondence between (BAF) β-complete and (SETAF) complete labelings, β-grounded and grounded labelings, β-preferred and preferred labelings, β-stable and stable labelings, β-semi-stable and semi-stable labelings. Besides semantic equivalences, we also show structural (or syntactic) equivalences between BAFs and SETAFs by finding subsets of them for which the proposed translations are each other's inverse up to isomorphism.

Parts of this chapter appeared in the work of Cordeiro; Alcântara (2026).

5.1 Introduction

From the connections between *BAF*s (Definition 15) and *NLP*s (Definition 19) in Chapter 4 and existing links between *NLP*s and *SETAF*s (Definition 16) from the works of König; Rapberger; Ulbricht (2022); Alcântara; Cordeiro; Sá (2024), we can find an indirect path connecting *BAF*s and *SETAF*s via logic programming. However, there remain questions about whether these connections can be naturally made more direct, and for which class of formalisms these translations can be made invertible up to isomorphism, i.e., preserving attack/support relation structures while disregarding argument names.

We study correspondences for the following semantics:

a) in *SETAF*s, the basic notion of a collective attack from a set of arguments S to an argument A is that A is rejected if every argument in S is accepted. This criterion gives rise to the labeling-based complete, grounded, preferred, stable, and semi-stable *SETAF* semantics (Definition 18);

b) in *BAF*s, attacks and supports between arguments can interact in many ways. For translating between *BAF*s and *SETAF*s, we consider the β-semantics introduced in Chapter 3. Recall that the basic notion of support provided by the β-semantics is that an argument A is rejected if every argument supporting A (including A

itself) is attacked by an accepted argument.

In both cases, argument A is rejected because of a *set* of accepted arguments. We consider the β-semantics (Definition 30) because its treatment of support aligns naturally with the notion of collective attacks in *SETAF*s (Definition 18). For finding structural correspondences, we take advantage of a fundamental result: mutually supporting arguments share the same label in the β-semantics (Proposition 43).

In Section 5.5, we show that other *BAF* semantics treat support cycles differently, and that this distinction makes directly applying our strategy to these semantics unviable. In our work, the correspondences we find involving *BAF*s are limited to the β-semantics, but similar strategies could help find equivalence results between other formalisms and/or under other semantics.

The main contributions of this chapter are twofold:

a) we provide direct translations from *BAF*s to *SETAF*s (and vice versa) and their semantics in a one-to-one correspondence between β-complete and complete labelings; β-grounded and grounded labelings; β-preferred and preferred labelings; β-stable and stable labelings; β-semi-stable and semi-stable labelings;
b) we find subclasses of *BAF*s and *SETAF*s for which the proposed translations satisfy syntactic equivalences, i.e., preserve attack/support relation structures.

This chapter is organized as follows: we show the translation from *BAF*s to *SETAF*s in Section 5.2, and from *SETAF*s to *BAF*s in Section 5.3; we discuss syntactic correspondences in Section 5.4; we compare our paper to related works in Section 5.5; conclusions are presented in Section 5.6.

5.2 From *BAF*s to *SETAF*s

In this section, we provide a translation from *BAF* $\mathcal{B}$ into its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$, such that there exists a one-to-one correspondence between β-complete, β-grounded, β-preferred, β-stable, β-semi-stable labelings of $\mathcal{B}$ and respectively complete, grounded, preferred, stable, and semi-stable labelings of $\mathfrak{A}_{\mathcal{B}}$.

5.2.1 SETAF construction

Recall that arguments with identical sets of supporters are labeled the same according to the β-semantics, i.e., for any β-complete labeling $\mathcal{L}$, it holds $\mathcal{L}(A) = \mathcal{L}(B)$ if $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ for arguments A and B (Proposition 43). For a more compact representation of arguments in the corresponding *SETAF*, arguments in *BAF* $\mathcal{B}$ with identical sets of supporters map to the same argument in the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$.

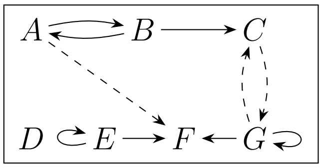


Figure 29 – Example *BAF* to illustrate the $\mathfrak{S}$-projection.

For instance, in the *BAF* from Figure 29, A maps to $\mathfrak{Sup}(A) = \{A\}$, B maps to $\mathfrak{Sup}(B) = \{B\}$, C and G map to $\mathfrak{Sup}(C) = \mathfrak{Sup}(G) = \{C, G\}$, D maps to $\mathfrak{Sup}(D) = \{D\}$, E maps to $\mathfrak{Sup}(E) = \{E\}$, and F maps to $\mathfrak{Sup}(F) = \{A, F\}$. We introduce a convenient notation for this mapping:

Definition 47 ($\mathfrak{S}$-projection)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. For any set of arguments $\mathcal{S} \subseteq \mathcal{A}$, $\mathfrak{S}(\mathcal{S}) = \{\mathfrak{Sup}(X) \mid X \in \mathcal{S}\}$ is the set obtained by replacing each argument X in $\mathcal{S}$ by its set of supporters $\mathfrak{Sup}(X)$.

With this notation, the set of arguments $\mathcal{A}$ from a *BAF* maps to $\mathfrak{S}(\mathcal{A})$ in the corresponding *SETAF*. Now only the attack relation of the corresponding *SETAF* remains to be specified.

We will use plain letters (e.g., X) for elements of $\mathcal{A}$, caligraphic letters (e.g., $\mathcal{X}$) for elements of $2^{\mathcal{A}}$, and letters in boldface (e.g., **X**) for elements of $2^{2^{\mathcal{A}}}$. To clarify, each argument $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$ in the corresponding *SETAF* will be a set of arguments ($\mathcal{Y} \subseteq \mathcal{A}$) in the *BAF*. More specifically, the set of supporters of some argument $A \in \mathcal{A}$. Thus, we may write $\mathcal{Y} = \mathfrak{Sup}(A)$ for some $A \in \mathcal{A}$. In the corresponding *SETAF*, an attacker-set **X** of argument $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$ is a set $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$, i.e., **X** is a set of elements in $\mathfrak{S}(\mathcal{A})$, so each *element* of **X** is a set of arguments in $\mathcal{A}$.

Example 26. Consider again the *BAF* from Figure 29. Argument F has two supporters: A and F itself. In the corresponding *SETAF*, F will be represented by $\mathfrak{Sup}(F) = \{A, F\}$. We wish to find which sets of arguments in $\mathfrak{S}(\mathcal{A})$ should attack $\mathfrak{Sup}(F)$. According to the β-semantics,

F has two acceptance conditions: F is accepted iff (i) A only has rejected attackers or (ii) F only has rejected attackers. Equivalently, F is accepted iff (i) B is rejected, or (ii) E and G are rejected.

We can rewrite the disjunction (i) *or* (ii) as the conjunction (a) *and* (b) of the conditions below:

a) B or E are rejected;

b) B or G are rejected.

The conditions obtained from the β-semantics are rewritten to the format (a) and (b) by using the Tr operator (Definition 6): $\text{Tr}[\{\{B\},\{E,G\}\}] = \{\{B,E\},\{B,G\}\}$. Intuitively, $\mathcal{X}_1 = \{B,E\}$ and $\mathcal{X}_2 = \{B,G\}$ can be seen as attacker-sets of the *SETAF* argument $\mathfrak{Sup}(F)$ representing F, in the sense that for $\mathfrak{Sup}(F)$ to be accepted, both $\mathcal{X}_1$ and $\mathcal{X}_2$ must contain some rejected argument. Thus, conditions (a) and (b) are more aligned to the notion of a collective attack in a *SETAF*. However, B, E, and G are arguments in $\mathcal{A}$ but not in $\mathfrak{S}(\mathcal{A})$, and any attacker-set of $\mathfrak{Sup}(F)$ should be a subset of $\mathfrak{S}(\mathcal{A})$ (and not of $\mathcal{A}$). The attacker-sets of $\mathfrak{Sup}(F)$ in this case are $\mathbf{X}'_1 = \{\mathfrak{Sup}(B),\mathfrak{Sup}(E)\} = \{\{B\},\{E\}\}$ and $\mathbf{X}'_2 = \{\mathfrak{Sup}(B),\mathfrak{Sup}(G)\} = \{\{B\},\{C,G\}\}$.

This strategy for determining the attacks in the corresponding *SETAF* is formalized next. Recall that $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$ is a subset of $\mathcal{A}$ and can be rewritten as $\mathfrak{Sup}(A)$ for some $A \in \mathcal{A}$.

Definition 48 (Corresponding *SETAF*)**.** Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*. The corresponding *SETAF* of $\mathcal{B}$ is $\mathfrak{A}_\mathcal{B} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_\mathcal{B})$, where $\mathit{Att}_\mathcal{B} \subseteq (2^{\mathfrak{S}(\mathcal{A})} - \emptyset) \times \mathfrak{S}(\mathcal{A})$ is a relation such that $(\mathbf{X},\mathcal{Y}) \in \mathit{Att}_\mathcal{B}$ iff $\mathbf{X} \in \text{Tr}[\{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathcal{Y}\}]$.

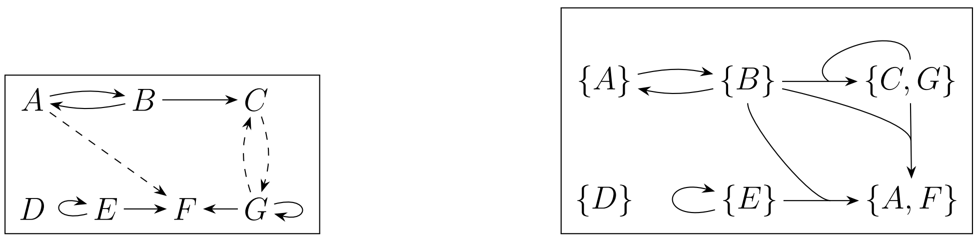
Example 27. For transforming the *BAF* $\mathcal{B}$ (Figure 30a) into its corresponding *SETAF* (Figure 30b), we follow the steps below:

a) compute $\mathfrak{S}(\mathcal{A})$ by finding $\mathfrak{Sup}(\cdot)$ for each argument of $\mathcal{B}$, i.e., $\mathfrak{Sup}(A) = \{A\}$, $\mathfrak{Sup}(B) = \{B\}$, $\mathfrak{Sup}(C) = \{C,G\}$, $\mathfrak{Sup}(D) = \{D\}$, $\mathfrak{Sup}(E) = \{E\}$, $\mathfrak{Sup}(F) = \{A,F\}$, and $\mathfrak{Sup}(G) = \{C,G\}$;

b) compute $\mathfrak{S}(\mathit{Att}(\cdot))$ for each argument of $\mathcal{B}$, i.e., $\mathfrak{S}(\mathit{Att}(A)) = \{\{B\}\}$, $\mathfrak{S}(\mathit{Att}(B)) = \{\{A\}\}$, $\mathfrak{S}(\mathit{Att}(C)) = \{\{B\}\}$, $\mathfrak{S}(\mathit{Att}(D)) = \emptyset$, $\mathfrak{S}(\mathit{Att}(E)) = \{\{E\}\}$, $\mathfrak{S}(\mathit{Att}(F)) = \{\{E\},\{C,G\}\}$, and $\mathfrak{S}(\mathit{Att}(G)) = \{\{C,G\}\}$;

c) for each argument $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$, find $\subseteq$-minimal sets $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(\mathit{Att}(Y)) \neq \emptyset$ for every $Y \in \mathcal{Y}$:

- for $\mathcal{Y} = \{A\}$, we have to find $\subseteq$-minimal $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(A)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \{\{B\}\} \neq \emptyset$. Hence, $\mathbf{X} = \{\{B\}\}$ is the only solution;
- for $\mathcal{Y} = \{B\}$, we have to find $\subseteq$-minimal $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(B)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \{\{A\}\} \neq \emptyset$. Hence, $\mathbf{X} = \{\{A\}\}$ is the only solution;
- for $\mathcal{Y} = \{C, G\}$, we have to find $\subseteq$-minimal $X \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(C)) \neq \emptyset$ and $\mathbf{X} \cap \mathfrak{S}(Att(G)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \{\{B\}\} \neq \emptyset$ and $\mathbf{X} \cap \{\{C, G\}\} \neq \emptyset$. Hence, $\mathbf{X} = \{\{B\}, \{C, G\}\}$ is the only solution;
- for $\mathcal{Y} = \{D\}$, we have to find $\subseteq$-minimal $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(D)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \emptyset \neq \emptyset$. There is no such $\mathbf{X}$;
- for $\mathcal{Y} = \{E\}$, we have to find $\subseteq$-minimal $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(E)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \{\{E\}\} \neq \emptyset$. Hence, $\mathbf{X} = \{\{E\}\}$ is the only solution;
- for $\mathcal{Y} = \{A, F\}$, we have to find $\subseteq$-minimal $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\mathbf{X} \cap \mathfrak{S}(Att(A)) \neq \emptyset$ and $\mathbf{X} \cap \mathfrak{S}(Att(F)) \neq \emptyset$, i.e., such that $\mathbf{X} \cap \{\{B\}\} \neq \emptyset$ and $\mathbf{X} \cap \{\{E\}, \{C, G\}\} \neq \emptyset$. Hence, $\mathbf{X} = \{\{B\}, \{E\}\}$ and $\mathbf{X} = \{\{B\}, \{C, G\}\}$ are the only solutions.



(a) $\mathcal{B}$ (b) $\mathfrak{A}_{\mathcal{B}}$

Figure 30 – *BAF* $\mathcal{B}$ from Example 27 and its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$.

This construction preserves the semantics of the original *BAF*, as shown in Table 6. In the sequel, we prove that this holds in general.

5.2.2 Equivalence results

We proceed by proving the equivalence between the semantics of a *BAF* $\mathcal{B}$ and their counterpart for the corresponding SETAF $\mathfrak{A}_{\mathcal{B}}$. For this purpose, we introduce two functions:

a) $\text{B2cS}_{\mathcal{B}\rightarrow\mathfrak{A}_{\mathcal{B}}}$ maps labelings of a *BAF* $\mathcal{B}$ to labelings of its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$. The acronym $\text{B2cS}_{\mathcal{B}\rightarrow\mathfrak{A}_{\mathcal{B}}}$ stands for *BAF* to corresponding *SETAF*;
b) $\text{cS2B}_{\mathcal{B}\rightarrow\mathfrak{A}_{\mathcal{B}}}$ maps labelings of the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ to labelings of the *BAF* $\mathcal{B}$. The acronym $\text{cS2B}_{\mathcal{B}\rightarrow\mathfrak{A}_{\mathcal{B}}}$ stands for corresponding *SETAF* to *BAF*.

β-complete labelings of $\mathcal{B}$	Complete labelings of $\mathfrak{A}_{\mathcal{B}}$
$\mathcal{L}_1 = (\{D\}, \emptyset, \{A, B, C, E, F, G\})$	$\mathcal{L}'_1 = (\{\{D\}\}, \emptyset, \{\{A\}, \{B\}, \{C, G\}, \{E\}, \{A, F\}\})$
$\mathcal{L}_2 = (\{B, D\}, \{A\}, \{C, E, F, G\})$	$\mathcal{L}'_2 = (\{\{B\}, \{D\}\}, \{\{A\}\}, \{\{C, G\}, \{E\}, \{A, F\}\})$
$\mathcal{L}_3 = (\{A, C, D, F, G\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{\{A\}, \{C, G\}, \{D\}, \{A, F\}\}, \{\{B\}\}, \{\{E\}\})$
β-grounded labelings of $\mathcal{B}$	Grounded labelings of $\mathfrak{A}_{\mathcal{B}}$
$\mathcal{L}_1 = (\{D\}, \emptyset, \{A, B, C, E, F, G\})$	$\mathcal{L}'_1 = (\{\{D\}\}, \emptyset, \{\{A\}, \{B\}, \{C, G\}, \{E\}, \{A, F\}\})$
β-preferred labelings of $\mathcal{B}$	Preferred labelings of $\mathfrak{A}_{\mathcal{B}}$
$\mathcal{L}_2 = (\{B, D\}, \{A\}, \{C, E, F, G\})$	$\mathcal{L}'_2 = (\{\{B\}, \{D\}\}, \{\{A\}\}, \{\{C, G\}, \{E\}, \{A, F\}\})$
$\mathcal{L}_3 = (\{A, C, D, F, G\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{\{A\}, \{C, G\}, \{D\}, \{A, F\}\}, \{\{B\}\}, \{\{E\}\})$
β-stable labelings of $\mathcal{B}$	Stable labelings of $\mathfrak{A}_{\mathcal{B}}$
None	None
β-semi-stable labelings of $\mathcal{B}$	Semi-stable labelings of $\mathfrak{A}_{\mathcal{B}}$
$\mathcal{L}_3 = (\{A, C, D, F, G\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{\{A\}, \{C, G\}, \{D\}, \{A, F\}\}, \{\{B\}\}, \{\{E\}\})$

Table 6 – Semantics for $\mathcal{B}$ and $\mathfrak{A}_{\mathcal{B}}$ from Example 27.

We then investigate the conditions under which these functions act as inverses of each other and employ these results to prove the equivalence between the semantics. Notably, these functions permit us to treat labelings of *BAF*s and of corresponding *SETAF*s interchangeably.

Definition 49 (B2cS$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$ and cS2B$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$ functions)**.** Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*, $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*, $\mathcal{L}ab_{\mathcal{B}}$ be the set of all labelings of $\mathcal{B}$, and $\mathcal{L}ab_{\mathfrak{A}_{\mathcal{B}}}$ be the set of all labelings of $\mathfrak{A}_{\mathcal{B}}$. We introduce a function B2cS$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}} : \mathcal{L}ab_{\mathcal{B}} \to \mathcal{L}ab_{\mathfrak{A}_{\mathcal{B}}}$ such that B2cS$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}) = \mathcal{L}'$, in which the following holds:

a) $\mathtt{in}(\mathcal{L}') = \{\mathcal{Y} \in \mathfrak{S}(\mathcal{A}) \mid \mathcal{L}(A') = \mathtt{in}$ for some $A' \in \mathcal{Y}\}$;
b) $\mathtt{out}(\mathcal{L}') = \{\mathcal{Y} \in \mathfrak{S}(\mathcal{A}) \mid \mathcal{L}(A') = \mathtt{out}$ for every $A' \in \mathcal{Y}\}$;
c) $\mathtt{undec}(\mathcal{L}') = \{\mathcal{Y} \in \mathfrak{S}(\mathcal{A}) \mid \mathcal{Y} \notin \mathtt{in}(\mathcal{L}') \cup \mathtt{out}(\mathcal{L}')\}$.

We introduce a function cS2B$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}} : \mathcal{L}ab_{\mathfrak{A}_{\mathcal{B}}} \to \mathcal{L}ab_{\mathcal{B}}$ such that cS2B$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}') = \mathcal{L}$, in which $\mathcal{L}(A) = \mathcal{L}'(\mathfrak{Sup}(A))$ for every $A \in \mathcal{A}$.

The direction from labelings of the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ to labelings of the *BAF* $\mathcal{B}$ is straightforward: the label assigned to an argument A in $\mathcal{B}$ follows from the label assigned to $\mathfrak{Sup}(A)$ in $\mathfrak{A}_{\mathcal{B}}$. In the inverse direction, an argument $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$ is accepted in $\mathfrak{A}_{\mathcal{B}}$ if some argument in $\mathcal{Y}$ is accepted in $\mathcal{B}$; is rejected in $\mathfrak{A}_{\mathcal{B}}$ if every argument in $\mathcal{Y}$ is rejected in $\mathcal{B}$; otherwise, $\mathcal{Y}$ is undecided in $\mathfrak{A}_{\mathcal{B}}$.

In general, cS2B$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$(B2cS$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$) is not equal to $\mathcal{L}$. For instance, considering the labeling $\mathcal{L} = (\{C\}, \{G\}, \{A, B, D, E, F\})$ of the *BAF* $\mathcal{B}$ of Figure 30a, we have cS2B$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$(B2cS$_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$) $= (\{C, G\}, \emptyset, \{A, B, D, E, F\})$. Such an inequality will always occur when two arguments with the same set of supporters receive distinct labels. In our example, arguments C and G have the same set of supporters $\{C, G\}$, but $\mathcal{L}(C) = \texttt{in}$ and $\mathcal{L}(G) = \texttt{out}$. It holds $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})(\{C, G\}) = \texttt{in}$ and thus both C and G are labeled in by $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}))$. However, for β-complete labelings, arguments with the same set of supporters have the same label (Proposition 43). A more general scenario in which the inequality arises is if $\mathcal{L}(A) = \texttt{in}$ and $\mathcal{L}(F) = \texttt{out}$, as $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})(\{A, F\}) = \texttt{in}$ and then both A and F are labeled in by $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}))$. This occurs because A is a supporter of F, but F received a smaller label w.r.t. to the ordering $\texttt{out} < \texttt{undec} < \texttt{in}$. In β-complete labelings, $A' \in \mathfrak{Sup}(A)$ implies $\mathcal{L}(A') \leq \mathcal{L}(A)$ w.r.t. to this ordering (this a direct consequence of Proposition 3). Thus, for β-complete labelings, we have that $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$:

Theorem 62. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_{\mathcal{B}})$ be its corresponding *SETAF*. For any β-complete labeling $\mathcal{L}$ of $\mathcal{B}$, it holds $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$.

In the reverse direction, $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}))$ is also not equal to $\mathcal{L}$, in general. For example, consider $\mathcal{L} = (\{\{A\}\}, \{\{A, F\}\}, \{\{B\}, \{C, G\}, \{D\}, \{E\}\})$ and the *SETAF* $\mathfrak{A}_{\mathcal{B}}$ of Figure 30b. We have $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = (\{\{A\}, \{A, F\}\}, \emptyset, \{\{B\}, \{C, G\}, \{D\}, \{E\}\})$. Such an inequality will always occur when argument $\mathcal{Y}$ receives a greater label than $\mathcal{Y}'$ (w.r.t. to the ordering $\texttt{out} < \texttt{undec} < \texttt{in}$) and $\mathcal{Y} \subseteq \mathcal{Y}'$. In our example, arguments $\{A\}$ and $\{A, F\}$ satisfy $\{A\} \subseteq \{A, F\}$, but $\mathcal{L}(\{A\}) = \texttt{in}$ and $\mathcal{L}(\{A, F\}) = \texttt{out}$. The reason for the inequality in this case is that $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})(A) = \texttt{in}$ and thus both $\{A\}$ and $\{A, F\}$ are labeled in by $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}))$. However, for complete labelings, we have that $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$:

Theorem 63. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_{\mathcal{B}})$ be its corresponding *SETAF*. For any complete labeling $\mathcal{L}$ of $\mathfrak{A}_{\mathcal{B}}$, it holds $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$.

Thus, when restricted to (β-)complete labelings, $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$ and $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$ are each other's inverse. From Theorems 62 and 63, we can obtain the following result:

Theorem 64. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$;

b) $\mathcal{L}$ is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}$.

Theorem 64 is one of the main results of this chapter. It plays a central role in ensuring the equivalence between the semantics for *BAF*s and their counterpart for *SETAF*s:

Theorem 65. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a grounded labeling of $\mathfrak{A}_{\mathcal{B}}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a preferred labeling of $\mathfrak{A}_{\mathcal{B}}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a stable labeling of $\mathfrak{A}_{\mathcal{B}}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a semi-stable labeling of $\mathfrak{A}_{\mathcal{B}}$.

The following result is a direct consequence of Theorems 63 and 65:

Corollary 66. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a grounded labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-grounded labeling of $\mathcal{B}$;

b) $\mathcal{L}$ is a preferred labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-preferred labeling of $\mathcal{B}$;

c) $\mathcal{L}$ is a stable labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-stable labeling of $\mathcal{B}$;

d) $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-semi-stable labeling of $\mathcal{B}$.

In the next section, we investigate the inverse direction, i.e., from *SETAF*s to *BAF*s.

5.3 From *SETAF*s to *BAF*s

Now we provide a translation from *SETAF*s to *BAF*s. As in the previous section, this translation guarantees the equivalence between the semantics for *SETAF*s and their counterpart for *BAF*s.

5.3.1 BAF construction

Given a *SETAF*, we obtain arguments and attacks in the corresponding *BAF* following Caminada; Sá, *et al.*'s (2015) work: each constructed argument is associated with a conclusion and a set of vulnerabilities, and an attack occurs whenever the conclusion of one argument appears among the vulnerabilities of another. Vulnerabilities of arguments and this strategy for determining the attack relation are also studied in the context of semi-structured argumentation (Rapberger; Ulbricht, 2023; Prakken, 2023). However, in our work, arguments remain abstract because conclusions and vulnerabilities are used only as intermediate steps for determining the attack and support relations, from which abstract *BAF* semantics then evaluate arguments.

According to the complete semantics, an argument A of a *SETAF* is accepted if every set attacking A has some rejected argument. In this case, we can construct a set of arguments $\mathcal{V}$ (standing for vulnerability) with only rejected attackers of A, which is a sufficient condition for accepting A in the β-complete semantics. By this approach, each argument of a *SETAF* (with possibly multiple vulnerabilities) will be represented by possibly multiple arguments supporting each other in the corresponding *BAF* (one for each vulnerability). The mutual support is used to link arguments in the corresponding *BAF* that correspond to the same argument in the *SETAF*. This is formalized as follows:

Definition 50 (Corresponding *BAF*)**.** Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF*. For any $A \in \mathcal{A}$, recall that $\mathit{Att}(A) \subseteq 2^{\mathcal{A}}$. We define the set of vulnerabilities of $A \in \mathcal{A}$ as $\mathbf{V}_A = \mathrm{Tr}[\mathit{Att}(A)]$. The corresponding *BAF* of $\mathfrak{A}$ is $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$, defined as follows:

a) $\mathcal{A}_{\mathfrak{A}} = \{(A, \mathcal{V}) \mid A \in \mathcal{A}, \mathcal{V} \in \mathbf{V}_A\}$;

b) $\mathit{Att}_{\mathfrak{A}} = \{((A, \mathcal{V}), (A', \mathcal{V}')) \in \mathcal{A}_{\mathfrak{A}} \times \mathcal{A}_{\mathfrak{A}} \mid A \in \mathcal{V}'\}$;

c) $\mathit{Sup}_{\mathfrak{A}} = \{((A, \mathcal{V}), (A, \mathcal{V}')) \in \mathcal{A}_{\mathfrak{A}} \times \mathcal{A}_{\mathfrak{A}} \mid \mathcal{V} \neq \mathcal{V}'\}$.

The main difficulty in computing the corresponding *BAF* is finding the set of vulnerabilities for each argument. In constrast, the attack and support relations follow directly from the set of arguments in the corresponding *BAF*.

Example 28. For transforming the *SETAF* $\mathfrak{A}$ (Figure 31a) into its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ (Figure 31b), we follow the steps below:

a) compute $\mathcal{A}_{\mathfrak{A}}$ by finding $\mathbf{V}_A$ for each argument A of $\mathcal{B}$:

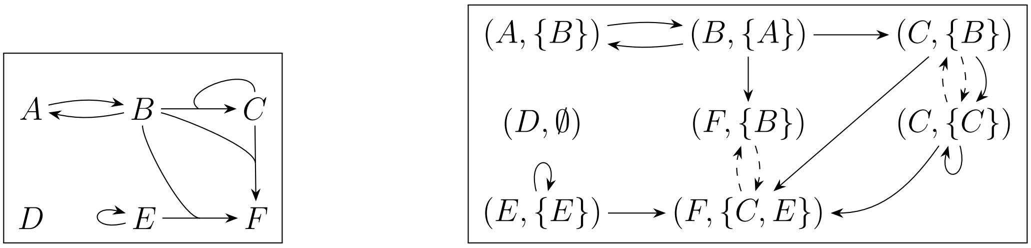
- for A, we find $\subseteq$-minimal $\mathcal{V} \subseteq \mathcal{A}$ such that $\{B\} \cap \mathcal{V} \neq \emptyset$. Hence, $\mathcal{V} = \{B\}$ is the only solution;
- for B, we find $\subseteq$-minimal $\mathcal{V} \subseteq \mathcal{A}$ such that $\{A\} \cap \mathcal{V} \neq \emptyset$. Hence, $\mathcal{V} = \{A\}$ is the only solution;
- for C, we find $\subseteq$-minimal $\mathcal{V} \subseteq \mathcal{A}$ such that $\{B, C\} \cap \mathcal{V} \neq \emptyset$. Hence, $\mathcal{V} = \{B\}$ and $\mathcal{V} = \{C\}$ are the only solutions;
- for D, we find that any $\mathcal{V} \subseteq \mathcal{A}$ satisfies $X \cap \mathcal{V} \neq \emptyset$ for every $X \in \mathit{Att}(D)$, as $\mathit{Att}(D) = \emptyset$. Hence, $\mathcal{V} = \emptyset$ is the only $\subseteq$-minimal solution;

– for E, we find $\subseteq$-minimal $\mathcal{V} \subseteq \mathcal{A}$ such that $\{E\} \cap \mathcal{V} \neq \emptyset$. Hence, $\mathcal{V} = \{E\}$ is the only solution;

– for F, we find $\subseteq$-minimal $\mathcal{V} \subseteq \mathcal{A}$ such that $\{B, E\} \cap \mathcal{V} \neq \emptyset$ and $\{B, C\} \cap \mathcal{V} \neq \emptyset$. Hence, $\mathcal{V} = \{B\}$ and $\mathcal{V} = \{C, E\}$ are the only solutions.

Then, $\mathcal{A}_{\mathfrak{A}} = \{(A, \{B\}), (B, \{A\}), (C, \{B\}), (C, \{C\}), (D, \emptyset), (E, \{E\}), (F, \{B\}), (F, \{C, E\})\}$;

b) *Att*$_{\mathfrak{A}}$ and *Sup*$_{\mathfrak{A}}$ are then easily obtained from $\mathcal{A}_{\mathfrak{A}}$.



(a) $\mathfrak{A}$ (b) $\mathcal{B}_{\mathfrak{A}}$

Figure 31 – *SETAF* $\mathfrak{A}$ from Example 28 and its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$

This construction preserves the semantics of the original *SETAF* from Example 28, as shown in Table 7. In the sequel, we prove that this holds in general.

Complete labelings of $\mathfrak{A}$	β-complete labelings of $\mathcal{B}_{\mathfrak{A}}$
$\mathcal{L}_1 = (\{D\}, \emptyset, \{A, B, C, E, F\})$	$\mathcal{L}'_1 = (\{D_1\}, \emptyset, \{A_1, B_1, C_1, C_2, E_1, F_1, F_2\})$
$\mathcal{L}_2 = (\{B, D\}, \{A\}, \{C, E, F\})$	$\mathcal{L}'_2 = (\{B_1, D_1\}, \{A_1\}, \{C_1, C_2, E_1, F_1, F_2\})$
$\mathcal{L}_3 = (\{A, C, D, F\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{A_1, C_1, C_2, D_1, F_1, F_2\}, \{B_1\}, \{E_1\})$
Grounded labelings of $\mathfrak{A}$	β-grounded labelings of $\mathcal{B}_{\mathfrak{A}}$
$\mathcal{L}_1 = (\{D\}, \emptyset, \{A, B, C, E, F\})$	$\mathcal{L}'_1 = (\{D_1\}, \emptyset, \{A_1, B_1, C_1, C_2, E_1, F_1, F_2\})$
Preferred labelings of $\mathfrak{A}$	β-preferred labelings of $\mathcal{B}_{\mathfrak{A}}$
$\mathcal{L}_2 = (\{B, D\}, \{A\}, \{C, E, F\})$	$\mathcal{L}'_2 = (\{B_1, D_1\}, \{A_1\}, \{C_1, C_2, E_1, F_1, F_2\})$
$\mathcal{L}_3 = (\{A, C, D, F\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{A_1, C_1, C_2, D_1, F_1, F_2\}, \{B_1\}, \{E_1\})$
Stable labelings of $\mathfrak{A}$	β-stable labelings of $\mathcal{B}_{\mathfrak{A}}$
None	None
Semi-stable labelings of $\mathfrak{A}$	β-semi-stable labelings of $\mathcal{B}_{\mathfrak{A}}$
$\mathcal{L}_3 = (\{A, C, D, F\}, \{B\}, \{E\})$	$\mathcal{L}'_3 = (\{A_1, C_1, C_2, D_1, F_1, F_2\}, \{B_1\}, \{E_1\})$

Table 7 – Semantics for $\mathfrak{A}$ and $\mathcal{B}_{\mathfrak{A}}$ from Example 28, where $A_1 = (A, \{B\}), B_1 = (B, \{A\}), C_1 = (C, \{B\}), C_2 = (C, \{C\}), D_1 = (D, \emptyset), E_1 = (E, \{E\}), F_1 = (F, \{B\})$, and $F_2 = (F, \{C, E\})$.

5.3.2 Equivalence results

We proceed by proving the equivalence between the semantics of *SETAF* $\mathfrak{A}$ and their counterpart for the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$. As in Section 5.2.2, we introduce two functions:

a) $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ maps labelings of a *SETAF* $\mathfrak{A}$ to labelings of its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$. The acronym $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ stands for *SETAF* to corresponding *BAF*;

b) $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ maps labelings of the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ to labelings of the *SETAF* $\mathfrak{A}$. The acronym $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ stands for corresponding *BAF* to *SETAF*.

We then investigate under which conditions these functions act as inverses of each other and employ these results to prove the equivalence between the semantics. Notably, these functions permit us to treat labelings of *SETAF*s and of corresponding *BAF*s interchangeably.

Definition 51 ($\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ and $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ functions)**.** Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF*, $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*, $\mathcal{L}ab_{\mathfrak{A}}$ be the set of all labelings of $\mathfrak{A}$, and $\mathcal{L}ab_{\mathcal{B}_{\mathfrak{A}}}$ be the set of all labelings of $\mathcal{B}_{\mathfrak{A}}$. We introduce a function $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}} : \mathcal{L}ab_{\mathfrak{A}} \to \mathcal{L}ab_{\mathcal{B}_{\mathfrak{A}}}$ such that $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})((A, \mathcal{V})) = \mathcal{L}(A)$ for any $(A, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$, and we define $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}} : \mathcal{L}ab_{\mathcal{B}_{\mathfrak{A}}} \to \mathcal{L}ab_{\mathfrak{A}}$ such that $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}) = \mathcal{L}'$, in which the following holds:

a) $\mathtt{in}(\mathcal{L}') = \{A \in \mathcal{A} \mid \mathcal{L}((A, \mathcal{V})) = \mathtt{in} \text{ for some } \mathcal{V} \in \mathbf{V}_A\}$;

b) $\mathtt{out}(\mathcal{L}') = \{A \in \mathcal{A} \mid \mathcal{L}((A, \mathcal{V})) = \mathtt{out} \text{ for every } \mathcal{V} \in \mathbf{V}_A\}$;

c) $\mathtt{undec}(\mathcal{L}') = \{A \in \mathcal{A} \mid A \notin \mathtt{in}(\mathcal{L}') \cup \mathtt{out}(\mathcal{L}')\}$.

The direction from labelings of a *SETAF* $\mathfrak{A}$ to labelings of its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ is straightforward: the label assigned to an argument $(A, \mathcal{V})$ in $\mathcal{B}_{\mathfrak{A}}$ follows from the label assigned to A in $\mathfrak{A}$. In the other direction, an argument $A \in \mathfrak{A}$ is accepted in $\mathfrak{A}$ if $(A, \mathcal{V})$ is accepted in $\mathcal{B}_{\mathfrak{A}}$ for some vulnerability $\mathcal{V}$ of A; is rejected in $\mathfrak{A}$ if $(A, \mathcal{V})$ is rejected in $\mathcal{B}_{\mathfrak{A}}$ for every vulnerability $\mathcal{V}$ of A; otherwise, A is undecided in $\mathfrak{A}$. Differently from $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$ and $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}$, the function $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$ is a left inverse of $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}$:

Theorem 67. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. For any labeling $\mathcal{L}$ of $\mathfrak{A}$, it holds $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$.

However, in general, $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is not equal to $\mathcal{L}$. For instance, consider the labeling $\mathcal{L} = (\{(F, \{B\})\}, \{(F, \{C, E\})\}, \ldots)$ of the *BAF* $\mathcal{B}_{\mathfrak{A}}$ of Figure 31b, where $\mathtt{undec}(\mathcal{L}) = \mathcal{A}_{\mathfrak{A}} - (\mathtt{in}(\mathcal{L}) \cup \mathtt{out}(\mathcal{L}))$ is omitted for convenience. We have that $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}) = (\{F\}, \emptyset, \ldots)$ and $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = (\{(F, \{B\}), (F, \{C, E\})\}, \emptyset, \ldots)$. Such an inequality will always occur when the labels of $(A, \mathcal{V})$ and $(A, \mathcal{V}')$ differ for distinct vulnerabilities $\mathcal{V}, \mathcal{V}'$ of an argument A. In our example, $\texttt{in} = \mathcal{L}((F, \{B\})) \neq \mathcal{L}((F, \{C, E\})) = \texttt{out}$. It holds $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})(F) = \texttt{in}$ and thus both $(F, \{B\})$ and $(F, \{C, E\})$ are labeled in by $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$. However, as arguments $(A, \mathcal{V})$ and $(A, \mathcal{V}')$ mutually support each other, if $\mathcal{L}$ respects $\mathfrak{Sup}$, we always have $\mathcal{L}((A, \mathcal{V})) = \mathcal{L}((A, \mathcal{V}'))$. Thus, for labelings of $\mathcal{B}_{\mathfrak{A}}$ respecting $\mathfrak{Sup}$, we have $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$:

Theorem 68. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. For any labeling $\mathcal{L}$ of $\mathcal{B}_{\mathfrak{A}}$ respecting $\mathfrak{Sup}$, it holds $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$.

A similar result to Theorem 64 also holds:

Theorem 69. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ iff $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$;
b) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}$.

From Theorem 69, we can ensure the equivalence between the semantics for *SETAF*s and their counterpart for *BAF*s:

Theorem 70. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a grounded labeling of $\mathfrak{A}$ iff $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-grounded labeling of $\mathcal{B}_{\mathfrak{A}}$;
b) $\mathcal{L}$ is a preferred labeling of $\mathfrak{A}$ iff $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-preferred labeling of $\mathcal{B}_{\mathfrak{A}}$;
c) $\mathcal{L}$ is a stable labeling of $\mathfrak{A}$ iff $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-stable labeling of $\mathcal{B}_{\mathfrak{A}}$;
d) $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{A}$ iff $\mathrm{S2cB}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-semi-stable labeling of $\mathcal{B}_{\mathfrak{A}}$.

The following result is a direct consequence of Theorems 68 and 69:

Corollary 71. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a grounded labeling of $\mathfrak{A}$;
b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a preferred labeling of $\mathfrak{A}$;
c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a stable labeling of $\mathfrak{A}$;
d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\mathrm{cB2S}_{\mathfrak{A} \to \mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a semi-stable labeling of $\mathfrak{A}$.

In the next section, we show that the translation from a *BAF* to a *SETAF* (Definition 48) behaves as the inverse (up to isomorphism) of the translation from a *SETAF* to a *BAF* (Definition 50) for some classes of *BAF*s and *SETAF*s.

5.4 On the relation between *BAF*s and *SETAF*s

In preceding sections, we have shown how to translate *BAF*s to *SETAF*s, and vice versa, while preserving their corresponding semantics. Now we check under which conditions these translations are the inverses (up to isomorphism) of each other, showing that there are also syntactic correspondences between *BAF*s (under the β-semantics) and *SETAF*s. As in Chapter 4, we say *BAF*s $\mathcal{B} = (\mathcal{A}, Att, Sup)$ and $\mathcal{B}' = (\mathcal{A}', Att', Sup')$ are isomorphic if there exists a bijection $f : \mathcal{A} \to \mathcal{A}'$ satisfying both conditions below:

a) $\{(f(A), f(B)) \mid (A, B) \in Att\} = Att'$;

b) $\{(f(A), f(B)) \mid (A, B) \in Sup\} = Sup'$.

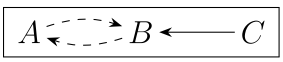
Similarly, *SETAF*s $\mathfrak{A} = (\mathcal{A}, Att)$ and $\mathfrak{A}' = (\mathcal{A}', Att')$ are isomorphic if there exists a bijection $f : \mathcal{A} \to \mathcal{A}'$ such that $\{f(X) \mid X \in \mathcal{X}\} \in Att'(f(A))$ iff $\mathcal{X} \in Att(A)$.

From a *BAF* $\mathcal{B}$, we obtain its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ via Definition 48; from $\mathfrak{A}_{\mathcal{B}}$, we obtain its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ via Definition 50. By following the other direction, from a *SETAF* $\mathfrak{A}$, we obtain its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$, and from $\mathcal{B}_{\mathfrak{A}}$, its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$. In this section, we investigate for which class of *BAF*s we have $\mathcal{B}$ is isomorphic to $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$, and for which class of *SETAF*s we have $\mathfrak{A}$ is isomorphic to $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$.

5.4.1 On the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$

Initially, we show that, in general, $\mathcal{B}$ is not isomorphic to $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$.

Example 29. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be the following *BAF*:



The corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ has no attacks and 2 arguments: $\{A, B\}$ and $\{C\}$. The corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ has 2 arguments: $(\{A, B\}, \emptyset)$ and $(\{C\}, \emptyset)$. There are no attacks or supports in $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$. Clearly, $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are not isomorphic.

This happened because, in $\mathcal{B}$, we have $A \in \mathfrak{Sup}(B)$ and $Att(A) \subset Att(B)$, whereas the following property is satisfied for any corresponding *BAF*:

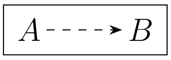
Proposition 72. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. For any $X \in \mathcal{A}_{\mathfrak{A}}$, there is no $X' \in \mathfrak{Sup}_{\mathfrak{A}}(X)$ such that $Att_{\mathfrak{A}}(X') \subset Att_{\mathfrak{A}}(X)$.

Proposition 72 shows that any corresponding *BAF* must necessarily have only minimal attacks. This motivates the definition of a *BAF* of minimal attacks:

Definition 52 (*BAF* of minimal attacks)**.** We say a *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$ is of minimal attacks if for any $X \in \mathcal{A}$, there is no $X' \in \mathfrak{Sup}(X)$ such that $Att(X') \subset Att(X)$.

However, this is not the only decisive factor for determining whether $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are isomorphic.

Example 30. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be the *BAF* shown below:

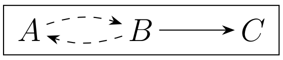


The corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ has no attacks and 2 arguments: $\{A\}$ and $\{A, B\}$. The corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ has 2 arguments: $(\{A\}, \emptyset)$ and $(\{A, B\}, \emptyset)$. There are no attacks or supports in $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$. Clearly, $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ have different number of supports and are not isomorphic.

This happened because, in $\mathcal{B}$, we have $A \in \mathfrak{Sup}(B)$ but $\mathfrak{Sup}(A) \neq \mathfrak{Sup}(B)$, whereas $\mathfrak{Sup}(A)$ and $\mathfrak{Sup}(B)$ would coincide in a corresponding *BAF*, given $A \in \mathfrak{Sup}(B)$. This happens because any corresponding *BAF* is a $\mathfrak{S}$-*BAF* (Definition 40, Lemma 49).

We now analyze another scenario.

Example 31. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be the following *BAF*:



The corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ has 2 arguments: $\{A, B\}$ and $\{C\}$. It has an attack from the singleton set $\{\{A, B\}\}$ (with only 1 argument $\{A, B\}$) towards argument $\{C\}$. The corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ has 2 arguments: $(\{A, B\}, \emptyset)$ and $(\{C\}, \{\{A, B\}\})$. In $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$, there is only one attack from argument $(\{A, B\}, \emptyset)$ towards argument $(\{C\}, \{\{A, B\}\})$, and there are no supports. We remark that, incidentally, *SETAF* $\mathfrak{A}_{\mathcal{B}}$ and *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ amount to standard *AAF*s. Clearly, $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ have different number of arguments and are not isomorphic.

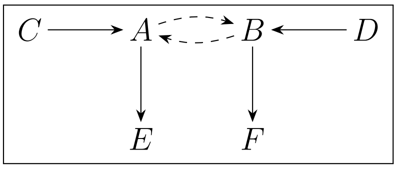
This happened because, in $\mathcal{B}$, we have $A \neq B$ and $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and $Att(A) = Att(B)$. Such arguments are redundant and cannot occur in a corresponding *BAF*. We say such *BAF*s are redundancy-free, i.e., *RFBAF*s (Definition 41). We recall from Chapter 4 that, for convenience, when a *BAF* is both a $\mathfrak{S}$-*BAF* and a *RFBAF*, we say it is a $\mathfrak{S}$-*RFBAF*.

By Definition 50, any corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*RFBAF*.

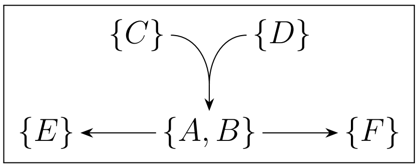
Proposition 73. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. It holds $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*RFBAF*.

This is useful, because it means that the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ can only hold if $\mathcal{B}$ is a $\mathfrak{S}$-*RFBAF*. Next, we discuss one more criterion necessary for the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$.

Example 32. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be the following *BAF*:

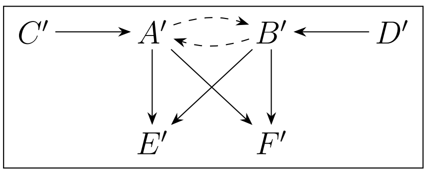


The corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ is shown below:



It has 5 arguments: $\{A, B\}$, $\{C\}$, $\{D\}$, $\{E\}$, and $\{F\}$. There is a collective attack from $\{\{C\}, \{D\}\}$ (two arguments) towards $\{A, B\}$ (1 argument). For the other attacks, the attacker-set is a singleton: $\{\{A, B\}\}$ is the only attacker-set of $\{E\}$ and $\{F\}$.

The corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ is shown below:



It has 6 arguments: $A' = (\{A, B\}, \{\{C\}\})$, $B' = (\{A, B\}, \{\{D\}\})$, $C' = (\{C\}, \emptyset)$, $D' = (\{D\}, \emptyset)$, $E' = (\{E\}, \{\{A, B\}\})$, and $F' = (\{F\}, \{\{A, B\}\})$. The set notation can be confusing, specially when computing minimal transversals, so we will show how to obtain these arguments in detail:

a) let's compute $\text{Tr}[Att_{\mathfrak{A}_{\mathcal{B}}}(\{A,B\})]$. In the *SETAF* $\mathfrak{A}_{\mathcal{B}}$, the argument $\{A,B\}$ has one attacker-set (with two arguments): $\{\{C\},\{D\}\}$. Let's denote by X the attacker-set $\{\{C\},\{D\}\}$. We have $Att_{\mathfrak{A}_{\mathcal{B}}}(\{A,B\}) = \{X\}$ is a singleton. A minimal transversal T of $\{X\}$ is a minimal set T satisfying $T \cap X \neq \emptyset$. Specifically, $T \cap \{\{C\},\{D\}\} \neq \emptyset$. We find two minimal T satisfying this condition: $T_0 = \{\{C\}\}$ and $T_1 = \{\{D\}\}$. Hence, in the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ we have arguments $(\{A,B\},T_0)$ and $(\{A,B\},T_1)$;

b) let's compute $\text{Tr}[Att_{\mathfrak{A}_{\mathcal{B}}}(\{E\})]$. In the *SETAF* $\mathfrak{A}_{\mathcal{B}}$, argument $\{E\}$ has one attacker-set (with one argument): $\{\{A,B\}\}$. Let's denote the attacker-set $\{\{A,B\}\}$ by X. We have $Att_{\mathfrak{A}_{\mathcal{B}}}(\{E\}) = \{X\}$ is a singleton. A minimal tranversal T of $\{X\}$ is a minimal set satisfying $T \cap X \neq \emptyset$. Specifically, $T \cap \{\{A,B\}\} \neq \emptyset$. We find only one minimal T satisfying this condition: $T = \{\{A,B\}\}$. Hence, in the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ we have an argument $(\{E\},\{\{A,B\}\})$;

c) similarly, we find an argument $(\{F\},\{\{A,B\}\})$ in the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$;

d) let's compute $\text{Tr}[Att_{\mathfrak{A}_{\mathcal{B}}}(\{C\})]$. In the *SETAF* $\mathfrak{A}_{\mathcal{B}}$, argument $\{C\}$ has no attacker-sets. We have $Att_{\mathfrak{A}_{\mathcal{B}}}(\{C\}) = \emptyset$. Also, $\text{Tr}[\emptyset] = \{\emptyset\}$, because any T satisfies $\forall H \in \emptyset, T \cap H \neq \emptyset$, and the empty set is the unique minimal T satisfying this condition. Hence, we find the argument $(\{C\},\emptyset)$ in the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$;

e) similarly, we find an argument $(\{D\},\emptyset)$.

We found 6 arguments in the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$. However, both arguments A' and B' attack the same arguments: namely, E' and F'. Hence, there are 6 attacks in $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$, whereas $\mathcal{B}$ only has 4 attacks. Clearly, $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are not isomorphic.

This happened because, in $\mathcal{B}$, we have $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, but A and B do not attack the same arguments. We recall the definition of a *BAF* of support-guided attacks, previously defined in Chapter 4 (Definition 42):

Definition 53 (*BAF* of support-guided attacks)**.** Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. If A and B attack the same arguments for any $A, B \in \mathcal{A}$ with $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, then we say $\mathcal{B}$ is of support-guided attacks.

By Definition 50, any corresponding *BAF* is of support-guided attacks.

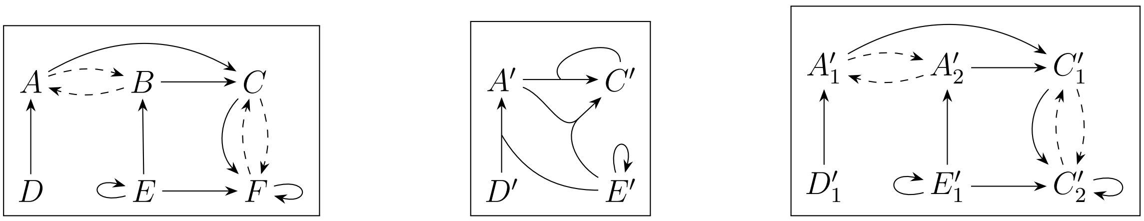
Proposition 74. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. It holds $\mathcal{B}_{\mathfrak{A}}$ is of support-guided attacks.

This is useful, because it means that the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ can only hold if $\mathcal{B}$ is a *BAF* of support-guided attacks. Now, we prove that the isomorphism holds precisely for the class of $\mathfrak{S}$-*RFBAF*s of minimal and support-guided attacks. For convenience, we will give this class a name.

Definition 54 ($\mathfrak{S}^*$-*RFBAF*)**.** Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. We say $\mathcal{B}$ is a $\mathfrak{S}^*$-*RFBAF* if $\mathcal{B}$ is a $\mathfrak{S}$-*RFBAF* of minimal and support-guided attacks.

Theorem 75. Let $\mathcal{B}$ be a *BAF*, $\mathfrak{A}_{\mathcal{B}}$ be its corresponding *SETAF*, and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ be the corresponding *BAF* of $\mathfrak{A}_{\mathcal{B}}$. It holds $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are isomorphic iff $\mathcal{B}$ is a $\mathfrak{S}^*$-*RFBAF*.

As expected from Theorem 75, *BAF*s $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ from Figure 32 are isomorphic, as $\mathcal{B}$ is a $\mathfrak{S}^*$-*RFBAF*.



(a) *BAF* $\mathcal{B}$ (b) Corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ (c) Corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$

Figure 32 – (a) *BAF* $\mathcal{B}$, (b) its corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ and (c) the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ of $\mathfrak{A}_{\mathcal{B}}$, where $A' = \{A, B\}$, $C' = \{C, F\}$, $D' = \{D\}$ and $E' = \{E\}$; $A'_1 = (A', \{D'\})$, $A'_2 = (A', \{E'\})$, $C'_1 = (C', \{A'\})$, $C'_2 = (C', \{C', E'\})$, $D'_1 = (D', \emptyset)$, and $E'_1 = (E', \{E'\})$.

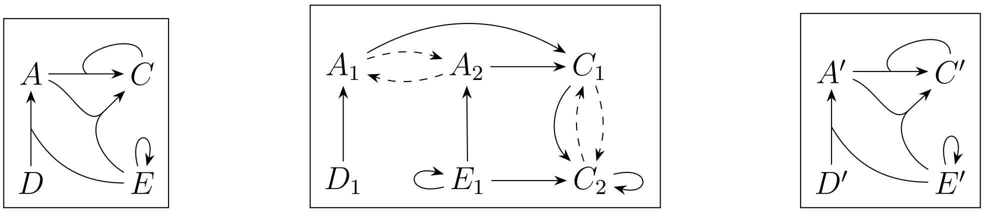
5.4.2 On the isomorphism between $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$

As opposed to the isomorphism between *BAF*s, we can ensure that each *SETAF* $\mathfrak{A}$ is isomorphic to $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ without any additional conditions on the *SETAF* $\mathfrak{A}$ (as in Definition 16). We remark that we add the minimality restriction to the original definition of *SETAF* from Nielsen; Parsons (2007) without changing the semantics considered in our work (Polberg, 2017; Dvořák; Rapberger; Woltran, 2020). However, the isomorphism between $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ only holds for frameworks without non-minimal attacks.

Theorem 76. Let $\mathfrak{A}$ be a *SETAF*, $\mathcal{B}_{\mathfrak{A}}$ be its corresponding *BAF*, and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ be the corresponding *SETAF* of $\mathcal{B}_{\mathfrak{A}}$. It holds $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ are isomorphic.

As expected from Theorem 76, *SETAF*s $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ of Figure 33 are isomorphic.

We conclude that when restricted to $\mathfrak{S}^*$-*RFBAF*s, the translation from *BAF*s to *SETAF*s (Definition 48) is the inverse up to isomorphism of the translation from *SETAF*s to *BAF*s (Definition 50). This means that *SETAF*s and $\mathfrak{S}^*$-*RFBAF*s have a deeper connection than *SETAF*s and *BAF*s, as in the former there is not only a semantic correspondence, but also a syntactic (or structural) one.



(a) *SETAF* $\mathfrak{A}$ (b) Corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ (c) Corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$

Figure 33 – (a) *SETAF* $\mathfrak{A}$, (b) its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ and (c) the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ of $\mathcal{B}_{\mathfrak{A}}$, where $A_1 = (A, \{D\})$, $A_2 = (A, \{E\})$, $C_1 = (C, \{A\})$, $C_2 = (C, \{C, E\})$, $D_1 = (D, \emptyset)$ and $E_1 = (E, \{E\})$; $A' = \{A_1, A_2\}$, $C' = \{C_1, C_2\}$, $D' = \{D_1\}$, and $E' = \{E_1\}$.

In the next section, we relate our findings in this chapter to existing approaches in the literature, and explain why our approach works for the β-semantics but fails for other *BAF* semantics.

5.5 Related works

In this section, we discuss the many existing connections between argumentation formalisms and relate them to our work. For instance, there are correspondences between *ADF*s and *SETAF*s (Polberg, 2017; Alcântara; Sá, 2021; Dvořák; Keshavarzi Zafarghandi; Woltran, 2023), *SETAF*s and *CAF*s (Dvořák; Rapberger; Woltran, 2020; Dvořák; Rapberger; Woltran, 2020; Rapberger, 2023), *ABA* frameworks and *SETAF*s (König; Rapberger; Ulbricht, 2022), *ABA* frameworks and *CAF*s (König; Rapberger; Ulbricht, 2022), *SETAF*s and *AAF*s (Flouris; Bikakis, 2019; Martin *et al.*, 2025). We discuss them in more detail:

a) Polberg (2017) translates *ADF*s to *SETAF*s, and *SETAF*s to *ADF*s — more specifically Attacking *ADF*s (ADF$^+$s) — in a one-to-one correspondence between

ADF extensions and *SETAF* extensions for the complete, grounded, preferred, and stable semantics;

b) Alcântara; Sá (2021) translate Attacking *ADF*s (ADF^+s) to *SETAF*s while preserving the complete, grounded, preferred, stable, and semi-stable semantics. They also demonstrate that this translation is the inverse of the one proposed by Polberg (2017) from *SETAF*s to *ADF*s when restricted to ADF^+s without redundant links. This means ADF^+s without redundant links and *SETAF*s are linked both semantically and syntactically;

c) Dvořák; Keshavarzi Zafarghandi; Woltran (2023) translate *SETAF*s to *ADF*s — more specifically, to *SETAF*-like *ADF*s (SETADFs). Then, SETADFs can be translated to *SETAF*s and also to the class of Support-Free *ADF*s (SFADFs). They find a one-to-one correspondence for *SETAF* labelings and *ADF* labelings for the complete, grounded, preferred, and stable semantics;

d) Rapberger (2023) translates well-formed *CAF*s to *SETAF*s (and vice versa) in a one-to-one syntactic correspondence and preserving the complete, grounded, preferred, and stable semantics. To also capture equivalences to the semi-stable semantics, they introduce *hybrid semantics* (h-semantics) for *CAF*s. In particular, the h-semi-stable semantics of *CAF*s corresponds to the semi-stable semantics of *SETAF*s.

e) Dvořák; Rapberger; Woltran (2020c, 2020a) map well-formed *CAF*s to *SETAF*s (and vice versa) while preserving the preferred, stable, and claim-based semi-stable semantics, but not the inherited semi-stable semantics;

f) König; Rapberger; Ulbricht (2022) translate *SETAF* to *ABA* frameworks (and vice versa) in a one-to-one correspondence between *SETAF* extensions and *ABA* extensions for the complete, grounded, preferred, and stable semantics. They also demonstrate that any *SETAF* $\mathfrak{A}$ coincides with the corresponding *SETAF* $\mathfrak{A}_{D_\mathfrak{A}}$ of the corresponding *ABA* $D_\mathfrak{A}$ of *SETAF* $\mathfrak{A}$;

g) König; Rapberger; Ulbricht (2022) translate well-formed *CAF*s to *ABA* frameworks and vice versa, in a one-to-one correspondence between *ABA* conclusion-extensions and *CAF* extensions for the complete, grounded, preferred, and stable semantics;

h) Flouris; Bikakis (2019) map *SETAF*s to *AAF*s, in a one-to-one correspondence between *SETAF* extensions and *AAF* extensions for the complete, grounded, preferred, and stable semantics. As each *AAF* can be naturally seen as a *SETAF*, the direction from *AAF*s to *SETAF*s is trivial.

i) Martin *et al.* (2025) study the relation between extension-based and labeling-based *node* and *arrow semantics* for *AAF*s and *SETAF*s. In particular, they show that *SETAF* arrow labelings correspond to node labelings of the corresponding *AAF* obtained via an *inside-out* translation, in which arrows of the *SETAF* become nodes of the corresponding *AAF*. Notably, the semi-stable semantics is preserved in this translation.

j) König; Rapberger; Ulbricht (2022) find semantic and syntactic correspondences between *NLP*s and *SETAF*s for the *SETAF* extension-based complete, grounded, preferred, and stable semantics. Alcântara; Cordeiro; Sá (2024) show this same connection from another perspective and showed it also preserves the semi-stable semantics.

We list some aspects in which these works differ from ours below:[1]

- a), c), f), and g) do not consider the semi-stable semantics;
- h) shows that their translation does not preserve the semi-stable semantics;
- a), c), e), h), and i) do not study syntactic correspondences (e.g., isomorphism results). For g), a result is shown for the corresponding *SETAF*s of *ABA* frameworks and *CAF*s, but not between *ABA* and *CAF* themselves;
- b), c), e), d), and g) do not encompass every instance of a formalism in some of the translation directions. Concretely, b) is restricted to ADF$^+$s; c), to SETADFs; d), e), and g), to well-formed *CAF*s. We recall that although our *isomorphism* results presented in Section 5.4 hold only for a subclass of *BAF*s (and any *SETAF*s), the *translations* defined in Sections 5.2 and 5.3 are applicable to any *BAF* and *SETAF*, respectively;
- f) finds a very strong syntactic correspondence for *SETAF*s (i.e., $\mathfrak{A} = \mathfrak{A}_{D_{\mathfrak{A}}}$), even stronger than an isomophism result like ours (Section 5.4). However, for *ABA*s, they present two translations to *SETAF*s, but do not study for which subclass of *ABA* these translations act as the inverse of the translation from *SETAF*s to *ABA*s.

In translations from well-formed *CAF*s to *SETAF*s (Dvořák; Rapberger; Woltran,

[1] We omit the letters from the standard ABNT notation for lists just this once, so as to avoid confusion with the list we are referencing.

2020; Dvořák; Rapberger; Woltran, 2020; Rapberger, 2023), each argument in the corresponding *SETAF* is a claim of an argument in the original *CAF*. This aligns with our approach of using the $\mathfrak{S}$-projection (Definition 47) to obtain the set of arguments $\mathfrak{S}(\mathcal{A})$ of the corresponding *SETAF* (Definition 48). The set of supporters $\mathfrak{Sup}(A)$ (Definition 24) of an argument A in a *BAF* takes the role of $cl(A)$ (the claim of A) in a *CAF*. This strategy allows us to obtain syntactic correspondences involving *SETAF*s and *BAF*s under the β-semantics (Section 5.4), which was similarly shown by Rapberger (2023) for well-formed *CAF*s. As the hybrid-semantics (Rapberger, 2023) and claim-based semantics (Dvořák; Rapberger; Woltran, 2020) for *CAF*s capture the semi-stable semantics for *SETAF*s, our work reveals that the β-semantics is a suitable interpretation of support for representing these (claim-based) notions in *BAF*s. Similarly, as König; Rapberger; Ulbricht (2022) found correspondences between *ABA* frameworks and *SETAF*s, our work evinces that *BAF*s under the β-semantics are a suitable target formalism for *ABA* instantiations.

In regards to *ADF*s, they are capable of encoding many kinds of interactions between arguments, so it is natural to expect they might generalize *BAF*s under the β-semantics. Our work confirms that this is true, because we show correspondences between *BAF*s (under the β-semantics) and *SETAF*s in both directions, and Alcântara; Sá (2021) proved that attacking *ADF*s (*ADF*$^+$s) are in semantic and syntactic correspondence with *SETAF*s in both directions.

By studying translations involving *BAF*s, our work serves as an additional motivation for the β-semantics (Chapter 3) among so many interpretations of support, as it is closely connected to well-known formalisms and their semantics. We now discuss why the equivalence results hold for the β-semantics, but do not directly extend to other *BAF* semantics.

Consider a *SETAF* $\mathfrak{A}_1$ with an argument A attacked by only two attacker-sets $\mathcal{X} = \{X_1, X_2\}$ and $\mathcal{Y} = \{Y_1, Y_2\}$ with $\mathcal{X} \cap \mathcal{Y} = \emptyset$. In a complete labeling $\mathcal{L}$ of $\mathfrak{A}_1$, the following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff (i) $\mathcal{L}(X) = \texttt{out}$ for some $X \in \mathcal{X}$ and (ii) $\mathcal{L}(Y) = \texttt{out}$ for some $Y \in \mathcal{Y}$;

b) $\mathcal{L}(A) = \texttt{out}$ iff (iii) $\mathcal{L}(X) = \texttt{in}$ for every $X \in \mathcal{X}$ or (iv) $\mathcal{L}(Y) = \texttt{in}$ for every $Y \in \mathcal{Y}$.

Argument A is accepted iff both (i) *and* (ii) are true. It is rejected iff (iii) *or* (iv) is true. We rewrite (i) and (ii) as (a) or (b) or (c) or (d) below:

a) $\mathcal{L}(X_1) = \texttt{out}$ and $\mathcal{L}(Y_1) = \texttt{out}$;

b) $\mathcal{L}(X_1) = \texttt{out}$ and $\mathcal{L}(Y_2) = \texttt{out}$;

c) $\mathcal{L}(X_2) = \texttt{out}$ and $\mathcal{L}(Y_1) = \texttt{out}$;

d) $\mathcal{L}(X_2) = \texttt{out}$ and $\mathcal{L}(Y_2) = \texttt{out}$.

The conditions above can be represented as the sets $\mathcal{V}_1 = \{X_1, Y_1\}$, $\mathcal{V}_2 = \{X_1, Y_2\}$, $\mathcal{V}_3 = \{X_2, Y_1\}$, and $\mathcal{V}_4 = \{X_2, Y_2\}$, each representing an acceptance condition for A. If we had found only one vulnerability set $\mathcal{V}$ for A, then we could define the attackers of A to be precisely *Att*$(A) = \mathcal{V}$ in the corresponding *BAF* and follow the usual interpretation from Dung's (1995) *AAF* semantics: A is accepted iff every argument in *Att*(A) is rejected; A is rejected iff some argument in *Att*(A) is accepted. Clearly, this does not work in general. For instance, A is accepted in $\mathfrak{A}_1$ when for *any* $\mathcal{V} \in \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4\}$, all arguments in $\mathcal{V}$ are rejected.

We find that the β-semantics is suitable for this representation. In our approach, the pairs $(A, \mathcal{V}_1)$, $(A, \mathcal{V}_2)$, $(A, \mathcal{V}_3)$, and $(A, \mathcal{V}_4)$ are arguments in the corresponding *BAF*, each representing one condition for the acceptance of A. The support relation connects pairs when they refer to the same argument (of the *SETAF*), which means these 4 arguments are in a support cycle. The strategy above of decomposing (i) and (ii) into a disjunction (a) or (b) or (c) or (d) is possible due to how support cycles are handled in the β-semantics. Proposition 43 guarantees that all arguments in the same support cycle share the same label in a β-complete labeling.

Contrastively, other works treat support cycles very differently. In some of them, *BAF*s are assumed to be acyclic with respect to the support relation (Amgoud *et al.*, 2008; Gottifredi *et al.*, 2018). In others, even when a support cycle is allowed, every argument in a support cycle is *always* rejected, because such cycles are seen as a form of circular reasoning (Nouioua; Risch, 2011; Nouioua; Boutouhami, 2023; Lagasquie-Schiex, 2023; Alfano *et al.*, 2024). Our strategy does not extend to these semantics, and it is an open question whether one can define syntactic correspondences between *BAF*s and *SETAF*s under these semantics.

5.6 Conclusions

In this chapter, we investigate the connections between *BAF*s under the β-semantics (Chapter 3) and *SETAF*s under the labeling-based acceptability semantics as characterized by Flouris; Bikakis (2019). We provide a mapping between *BAF*s and *SETAF*s (and vice versa), and also between their labelings, such that the β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable semantics for *BAF*s correspond to, respectively, the complete, grounded, preferred, stable, and semi-stable semantics for *SETAF*s.

Our translation from *SETAF*s to *BAF*s offers a key advantage over the translation from *SETAF*s to *AAF*s presented by Flouris; Bikakis (2019). In their approach, the semi-stable semantics for *SETAF*s do not correspond in general to the semi-stable semantics for *AAF*s, whereas our approach captures the equivalence between semi-stable labelings of *SETAF*s and β-semi-stable labelings of *BAF*s. Moreover, we guarantee additional results regarding the syntactic equivalence between *BAF*s and *SETAF*s. For any *SETAF* $\mathfrak{A}$ (with the minimality restraint as in Definition 16), we can obtain its corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ and the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ such that $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ are isomorphic. This means that, disregarding the names of arguments, from a corresponding *BAF* $\mathcal{B}'$, we can recover the original *SETAF* which originated $\mathcal{B}'$. In the other direction, from *BAF* $\mathcal{B}$ we obtain the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ and the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$. We find that the class of *BAF*s for which the isomorphism between $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ holds consists of redundancy-free *BAF*s of support cliques and of minimal and support-guided attacks ($\mathfrak{S}^*$-*RFBAF*s). This means that, disregarding the names of arguments, from a corresponding *SETAF* $\mathfrak{A}'$ originated from some $\mathfrak{S}^*$-*RFBAF*, we can recover the original *BAF* $\mathcal{B}$ which originated $\mathfrak{A}'$. Hence, when compared to the relationship between *SETAF*s and *AAF*s, the relationship between *SETAF*s and *BAF*s is demonstrably more robust. It extends beyond semantics to encompass structural aspects.

6 CONCLUSIONS

This dissertation contributes to a better understanding of how different notions in argumentation (such as support and collective attacks) relate to each other and to logic programming. We provide direct translations between *BAF*s, *SETAF*s, *NLP*s, and their semantics (including a new semantics for *BAF*s), and find classes for which these translations are invertible, showing that their connection is also structural. In this section, we recall these contributions, highlight limitations, and suggest future work.

In this work, we propose a new semantics for *BAF*s, called β-semantics, based on a dual characterization of defeat and defense. These semantics (namely β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable) generalize the corresponding *AAF* semantics and preserve fundamental results of *AAF*s, such as Dung's (1995) Fundamental Lemma.

Notably, the β-semantics is suitable for finding correspondences between *BAF*s and *NLP*s: we translate *NLP*s into *BAF*s and vice versa in a one-to-one correspondence between *NLP* partial stable, well-founded, regular, stable, and L-stable models and respectively *BAF* β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable labelings. Additionally, for the subclasses of Reduced Atomic Logic Programs (*RALP*s) and redundancy-free *BAF*s of support cliques and of support-guided attacks ($\mathfrak{S}^{+}$-*RFBAF*s), these translations act as each other's inverse (up to isomorphism):

a) from the corresponding *NLP* $P_{\mathcal{B}}$ of a $\mathfrak{S}^{+}$-*RFBAF* $\mathcal{B}$, we can obtain the corresponding *BAF* $\mathcal{B}_{P_{\mathcal{B}}}$, which is isomorphic to $\mathcal{B}$;
b) from the corresponding *BAF* $\mathcal{B}_P$ of an *RALP* P, we can obtain the corresponding *NLP* $P_{\mathcal{B}_P}$, which is isomorphic to P.

We also explain how to translate *NLP*s into *RALP*s and prove that both classes have the same expressiveness under the semantics studied in our work. This also suggests an alternative way of finding the corresponding *BAF* of an *NLP* P: instead of constructing arguments directly from P, we can first obtain the corresponding *RALP* P^* of P and then obtain arguments from P^*.

To extend these results for *SETAF*s, we propose translations from *SETAF*s to *BAF*s and vice versa in a one-to-one correspondence between *SETAF* complete, grounded, preferred, stable, and semi-stable labelings and respectively *BAF* β-complete, β-grounded, β-preferred, β-stable, and β-semi-stable labelings. Furthermore, for the subclass of *BAF*s of minimal and support-guided attacks ($\mathfrak{S}^{*}$-*RFBAF*s), these translations act as each other's inverse (up to isomorphism):

a) from the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}}$ of a $\mathfrak{S}^*$-*RFBAF* $\mathcal{B}$, we can obtain the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$, which is isomorphic to $\mathcal{B}$;

b) from the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$ of a *SETAF* $\mathfrak{A}$, we can obtain the corresponding *SETAF* $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$, which is isomorphic to $\mathfrak{A}$.

As a consequence of these findings, we conclude that *BAF*s, *SETAF*s, and *NLP*s are connected at both a semantic and syntactic level. By studying when these translations can be made invertible, we encounter and categorize many forms of redundancy. For *BAF*s, the subclasses $\mathfrak{S}^+$-*RFBAF*s and $\mathfrak{S}^*$-*RFBAF*s differ from each other in that the latter has an additional minimality constraint, mimicking the one from *SETAF*s (Definition 16). Similarly for *NLP*s, the difference between the subclasses *RALP*s (Chapter 4) and Redudancy-Free Atomic Logic Programs (*RFALP*s, König; Rapberger; Ulbricht 2022) (which have been shown to be in syntactic equivalence with *SETAF*s) is that *RFALP*s cannot have non-minimal rules, i.e., r, r' such that $body(r) \subset body(r')$ and $head(r) = head(r')$. Thus, $\mathfrak{S}^*$-*RFBAF*s and *RFALP*s — the more restrictive versions of respectively $\mathfrak{S}^+$-*RFBAF*s and *RALP*s — guarantee structural equivalences when relating *SETAF*s to *BAF*s and *NLP*s. The relationship between these classes is depicted in Figure 34, where blue lines indicate the subset relation and black arrows symbolize structural equivalence.

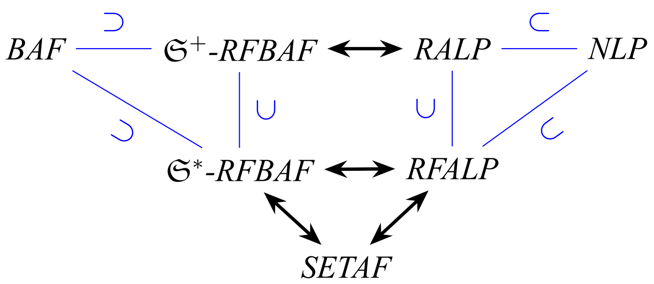


Figure 34 – Relationship between *SETAF*s, $\mathfrak{S}^*$-*RFBAF*s, $\mathfrak{S}^+$-*RFBAF*s, *BAF*s, *RFALP*s, *RALP*s, and *NLP*s.

From our results, we can expect that if we remove the minimality constraint in *SETAF*s (as in the original formulation by Nielsen; Parsons 2007), then this general class of *SETAF*s would be in syntactic correspondence with *RALP*s (on the *NLP* side) and $\mathfrak{S}^+$-*RFBAF*s (on the *BAF* side), both of which do not enforce a minimality constraint.

Regarding the significance and potential impact of these results, we highlight that pursuing this research direction yields insights into which forms of non-monotonic reasoning can and cannot be represented by formal argumentation. In particular, by enlightening these connections between *BAF*s, *SETAF*s, and *NLP*s, many approaches, semantics, and techniques naturally developed for one may be applied to the other, and vice versa. For instance, from an algorithm that transforms large *SETAF*s into smaller *SETAF*s we can derive an algorithm that transforms large *BAF*s to smaller *BAF*s via the translations presented in this paper. In addition, the representation of *NLP*s as argumentation frameworks offers an intuitive visualization of logic programs. Since *NLP*s are closely related to various other formalisms, some of them could benefit from insights derived from argumentation, specially given already known connections between *NLP*s, *ADF*s, *ABA* frameworks, and *CAF*s.

Despite the structural corrrespondences presented in our work, one limitation is that we study equivalences only by the perspective of intertranslatability. Another perspective is that of realizability, which compares formalisms/semantics by asking the question of "what kind of *problems* can the formalism solve?" (Strass, 2015; Pührer, 2020). For example, given a class $\mathcal{C}$ of an argumentation formalism and a semantics σ mapping each $C \in \mathcal{C}$ to a set of labelings $\mathbf{L}_C$ (where $\mathcal{L} \in \mathbf{L}_C$ is a function $\mathcal{L} : \mathcal{A} \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$ and $\mathcal{A}$ is the set of arguments of C), the signature $\Sigma_\sigma^{\mathcal{C}} = \{\sigma(C) \mid C \in \mathcal{C}\}$ represents every *answer* (interpretation) this combination of formalism and semantics can possibly give. In this sense, two classes $\mathcal{C}, \mathcal{C}'$ are equivalent according to a semantics σ when $\Sigma_\sigma^{\mathcal{C}} = \Sigma_\sigma^{\mathcal{C}'}$.

This also can be extended to compare distinct (but corresponding) semantics for $\subseteq$-incomparable classes of formalisms such as *SETAF*s and *BAF*s, in which neither is a subset of the other. For instance, let SETAF and BAF be the classes of all *SETAF*s and *BAF*s, then we can compare *SETAF* complete semantics σ_{co} with *BAF* β-complete semantics $\sigma_{\beta co}$ by checking whether $\Sigma_{\sigma_{co}}^{\text{SETAF}} = \Sigma_{\sigma_{\beta co}}^{\text{BAF}}$.

Our intertranslatability results show that the set of complete labelings $\sigma_{co}(\mathfrak{A})$ obtained from a *SETAF* $\mathfrak{A}$ is in one-to-one correspondence with the set of β-complete labelings $\sigma_{\beta co}(\mathcal{B}_{\mathfrak{A}})$ of the corresponding *BAF* $\mathcal{B}_{\mathfrak{A}}$. However, our translation changes the arguments in consideration: *SETAF* $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ maps to *BAF* $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ with $\mathcal{A} \neq \mathcal{A}_{\mathfrak{A}}$. Thus, given a *SETAF* $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$, we cannot guarantee the existence of a *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}', \mathit{Sup}')$ such that $\sigma_{co}(\mathfrak{A}) = \sigma_{\beta co}(\mathcal{B})$. Indeed, the equation $\Sigma_{\sigma_{co}}^{\text{SETAF}} = \Sigma_{\sigma_{\beta co}}^{\text{BAF}}$ as defined above does *not* hold, as shown by the next example.

Example 33. Consider the *SETAF* $\mathfrak{A}_1$ with arguments A, B, and C, depicted in Figure 35. Its complete labelings are $\mathcal{L}_1 = (\{A, B\}, \{C\}, \emptyset)$, $\mathcal{L}_2 = (\{A, C\}, \{B\}, \emptyset)$, $\mathcal{L}_3 = (\{B, C\}, \{A\}, \emptyset)$, and $\mathcal{L}_4 = (\emptyset, \emptyset, \{A, B, C\})$. By absurd, suppose a *BAF* $\mathcal{B} = (\{A, B, C\}, \mathit{Att}, \mathit{Sup})$ has β-

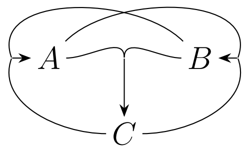


Figure 35 – *SETAF* $\mathfrak{A}_1$ from Example 33.

complete labelings coinciding with $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$. We conclude the following:

a) as $\texttt{in}(\mathcal{L}_1) = \{A, B\}$ and $\mathcal{L}_1(C) = \texttt{out}$, then $(A, C), (B, C) \notin$ *Sup*;

b) as $\texttt{in}(\mathcal{L}_2) = \{A, C\}$ and $\mathcal{L}_2(B) = \texttt{out}$, then $(A, B), (C, B) \notin$ *Sup*;

c) as $\texttt{in}(\mathcal{L}_3) = \{B, C\}$ and $\mathcal{L}_3(A) = \texttt{out}$, then $(B, A), (C, A) \notin$ *Sup*;

d) self-supports are ineffective in the β-semantics, hence we can consider that there are no self-supports in *Sup*;

e) given the items above, there are no supports in $\mathcal{B}$ (it amounts to a standard *AAF*);

f) as B is labeled $\texttt{in}$ by some β-complete labeling and there is no support, it follows B cannot attack itself;

g) as $\mathcal{L}_2(B) = \texttt{out}$, there is some attack targeting B (from either A or C);

h) if A attacks B, this contradicts $\mathcal{L}_1(A) = \texttt{in}$ and $\mathcal{L}_1(B) = \texttt{in}$;

i) if C attacks B, this contradicts $\mathcal{L}_3(C) = \texttt{in}$ and $\mathcal{L}_3(B) = \texttt{in}$.

Hence there is no solution *BAF* with β-complete labelings $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$.

This is due to the constraint of labelings being defined over the same set of arguments in both frameworks. Among *BAF*s, *SETAF*s, and *NLP*s, the capability of realizing sets of labelings can be further explored and compared.

Other future works include establishing semantic and syntactic equivalence for argumentation frameworks with other kinds of interactions between arguments, such as Argumentation Frameworks with Recursive Attacks (*AFRA*s) (Baroni *et al.*, 2011), in which attacks are themselves seen as defeasible entities susceptible to be the target of attacks.

REFERENCES

ALCÂNTARA, J.; CORDEIRO, R. Bipolar Argumentation Frameworks with a dual relation between defeat and defence. **Journal of Logic and Computation**, [Oxford], v. 35, n. 2, p. exae006, 2024. DOI: 10.1093/logcom/exae006.

ALCÂNTARA, J.; CORDEIRO, R. On the Equivalence between Logic Programs and Bipolar Argumentation Frameworks. **Journal of Artificial Intelligence Research**, [El Segundo], v. 84, n. 17, p. 1–43, 2025. DOI: 10.1613/jair.1.18086.

ALCÂNTARA, J.; CORDEIRO, R.; SÁ, S. On the equivalence between logic programming and SETAF. **Theory and Practice of Logic Programming**, [Cambridge], v. 24, n. 6, p. 1208–1236, 2024. DOI: 10.1017/S1471068424000188.

ALCÂNTARA, J.; SÁ, S. Equivalence results between SETAF and attacking abstract dialectical frameworks. *In:* INTERNATIONAL WORKSHOP ON NON-MONOTONIC REASONING, 19., 2021, Hanoi. **Proceedings** […]. [Aachen]: [CEUR-WS.org], 2021, p. 139–148.

ALCÂNTARA, J.; SÁ, S.; ACOSTA-GUADARRAMA, J. On the equivalence between abstract dialectical frameworks and logic programs. **Theory and Practice of Logic Programming**, [Cambridge], v. 19, n. 5/6, p. 941–956, 2019. DOI: 10.1017/S1471068419000280.

ALFANO, G. *et al.* Cyclic Supports in Recursive Bipolar Argumentation Frameworks: Semantics and LP Mapping. **Theory and Practice of Logic Programming**, [Cambridge], v. 24, n. 4, p. 921–941, 2024. DOI: 10.1017/S1471068424000310.

ALFANO, G. *et al.* On the semantics of abstract argumentation frameworks: A logic programming approach. **Theory and Practice of Logic Programming**, [Cambridge], v. 20, n. 5, p. 703–718, 2020. DOI: 10.1017/S1471068420000253.

AMGOUD, L. *et al.* On bipolarity in argumentation frameworks. **International Journal of Intelligent Systems**, [Hoboken, USA], v. 23, n. 10, p. 1062–1093, 2008. DOI: 10.1002/int.20307.

BARONI, P. *et al.* AFRA: Argumentation framework with recursive attacks. **International Journal of Approximate Reasoning**, [Amsterdam], v. 52, n. 1, p. 19–37, 2011. DOI: 10.1016/j.ijar.2010.05.004.

BERGE, C. **Hypergraphs: combinatorics of finite sets**. Amsterdam: Elsevier, 1984.

BOELLA, G. *et al.* Support in abstract argumentation. *In:* INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELS OF ARGUMENT, 3., 2010, Desenzano del Garda. **Proceedings** […]. Amsterdam: IOS Press, 2010, p. 111–122. DOI: 10.3233/978-1-60750-619-5-111.

BONDARENKO, A. *et al.* An abstract, argumentation-theoretic approach to default reasoning. **Artificial intelligence**, [Amsterdam], v. 93, n. 1/2, p. 63–101, 1997. DOI: 10.1016/S0004-3702(97)00015-5.

BREWKA, G.; WOLTRAN, S. Abstract dialectical frameworks. *In:* INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 12., 2010, Toronto, Ontario. **Proceedings** […]. [Washington, DC]: AAAI Press, 2010, p. 102–111.

CAMINADA, M. On the issue of reinstatement in argumentation. *In:* EUROPEAN CONFERENCE ON LOGICS IN ARTIFICIAL INTELLIGENCE, 10., 2006, Liverpool, UK. **Proceedings** […]. Berlin, Germany: Springer, 2006a, p. 111–123. DOI: 10.1007/11853886_11.

CAMINADA, M. Semi-stable semantics. *In:* INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELS OF ARGUMENT, 1., 2006, Liverpool, UK. **Proceedings** […]. Amsterdam: IOS Press, 2006b, p. 121–130.

CAMINADA, M.; AMGOUD, L. An axiomatic account of formal argumentation. *In:* NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 20., 2005, Pittsburgh, Pennsylvania. **Proceedings** […]. [Washington, DC]: AAAI Press, 2005, p. 608–613.

CAMINADA, M.; AMGOUD, L. On the evaluation of argumentation formalisms. **Artificial Intelligence**, [Amsterdam], v. 171, n. 5, p. 286–310, 2007. DOI: 10.1016/j.artint.2007.02.003.

CAMINADA, M.; GABBAY, D. M. A logical account of formal argumentation. **Studia Logica**, [Amsterdam], v. 93, p. 109–145, 2009. DOI: 10.1007/s11225-009-9218-x.

CAMINADA, M.; HARIKRISHNAN, S.; SÁ, S. Comparing logic programming and formal argumentation; the case of ideal and eager semantics. **Argument & Computation**, [London, England], v. 13, n. 1, p. 93–120, 2022. DOI: 10.3233/AAC-200528.

CAMINADA, M.; SÁ, S., *et al.* On the equivalence between logic programming semantics and argumentation semantics. **International Journal of Approximate Reasoning**, [Amsterdam], v. 58, p. 87–111, 2015. DOI: 10.1016/j.ijar.2014.12.004.

CAMINADA, M.; SCHULZ, C. On the equivalence between assumption-based argumentation and logic programming. **Journal of Artificial Intelligence Research**, [El Segundo], v. 60, p. 779–825, 2017. DOI: 10.1613/jair.5581.

CAYROL, C.; LAGASQUIE-SCHIEX, M.-C. Bipolarity in argumentation graphs: Towards a better understanding. **International Journal of Approximate Reasoning**, [Amsterdam], v. 54, n. 7, p. 876–899, 2013. DOI: 10.1016/j.ijar.2013.03.001.

CAYROL, C.; LAGASQUIE-SCHIEX, M.-C. Coalitions of arguments in bipolar argumentation frameworks. *In:* INTERNATIONAL WORKSHOP ON COMPUTATIONAL

MODELS OF NATURAL ARGUMENT, 7., 2007, Hyderabad, India. **Proceedings** [...]. [*S. l.*]: [*s. n.*], 2007, p. 14–23.

CAYROL, C.; LAGASQUIE-SCHIEX, M.-C. On the acceptability of arguments in bipolar argumentation frameworks. *In:* EUROPEAN CONFERENCE ON SYMBOLIC AND QUANTITATIVE APPROACHES TO REASONING AND UNCERTAINTY, 8., 2005, Barcelona, Spain. **Proceedings** [...]. Berlin, Germany: Springer, 2005, p. 378–389. DOI: 10.1007/11518655_33.

COHEN, A. *et al.* A survey of different approaches to support in argumentation systems. **The Knowledge Engineering Review**, [Cambridge], v. 29, n. 5, p. 513–550, 2014. DOI: 10.1017/S0269888913000325.

CORDEIRO, R.; ALCÂNTARA, J. On the Equivalence between Bipolar Argumentation Frameworks and SETAFs. **Argument & Computation**, [Amsterdam], 2026. In press.

ČYRAS, K.; SCHULZ, C.; TONI, F. Capturing bipolar argumentation in non-flat assumption-based argumentation. *In:* INTERNATIONAL CONFERENCE ON PRINCIPLES AND PRACTICE OF MULTI-AGENT SYSTEMS, 20., 2017, Nice, France. **Proceedings** [...]. Cham, Switzerland: Springer, 2017, p. 386–402. DOI: 10.1007/978-3-319-69131-2_23.

DUNG, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. **Artificial intelligence**, [Amsterdam], v. 77, n. 2, p. 321–357, 1995. DOI: 10.1016/0004-3702(94)00041-X.

DUNNE, P. E. *et al.* Characteristics of multiple viewpoints in abstract argumentation. **Artificial Intelligence**, [Amsterdam], v. 228, p. 153–178, 2015. DOI: 10.1016/j.artint.2015.07.006.

DVOŘÁK, W.; GAGGL, S. A., *et al.* Making use of advances in answer-set programming for abstract argumentation systems. *In:* INTERNATIONAL CONFERENCE ON APPLICATIONS OF DECLARATIVE PROGRAMMING AND KNOWLEDGE MANAGEMENT AND WORKSHOP ON LOGIC PROGRAMMING, 19;25., 2011, Vienna, Austria. **Proceedings** [...]. Berlin, Germany: Springer, 2013, p. 114–133. DOI: 10.1007/978-3-642-41524-1_7.

DVOŘÁK, W.; KESHAVARZI ZAFARGHANDI, A.; WOLTRAN, S. Expressiveness of SETAFs and support-free ADFs under 3-valued semantics. **Journal of Applied Non-Classical Logics**, [London, UK], v. 33, n. 3/4, p. 298–327, 2023. DOI: 10.1080/11663081.2023.2244361.

DVOŘÁK, W.; RAPBERGER, A.; WOLTRAN, S. Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to SETAFs. *In:* INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 17., 2020, Rhodes, Greece (Online). **Proceedings** [...]. [*S. l.*]: IJCAI, 2020a, p. 341–350. DOI: 10.24963/kr.2020/35.

DVOŘÁK, W.; RAPBERGER, A.; WOLTRAN, S. On the different types of collective attacks in abstract argumentation: equivalence results for SETAFs. **Journal of Logic and Computation**, [Oxford], v. 30, n. 5, p. 1063–1107, 2020b. DOI: 10.1093/logcom/exaa033.

DVOŘÁK, W.; RAPBERGER, A.; WOLTRAN, S. On the relation between claim-augmented argumentation frameworks and collective attacks. *In:* EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 24., 2020, Santiago de Compostela. **Proceedings** […]. Amsterdam: IOS Press, 2020c, p. 721–728. DOI: 10.3233/FAIA200159.

DVOŘÁK, W.; WOLTRAN, S. Complexity of abstract argumentation under a claim-centric view. **Artificial Intelligence**, [Amsterdam], v. 285, p. 103290, 2020. DOI: 10.1016/j.artint.2020.103290.

EITER, T.; LEONE, N.; SACCA, D. On the partial semantics for disjunctive deductive databases. **Annals of Mathematics and Artificial Intelligence**, [Basel], v. 19, p. 59–96, 1997. DOI: 10.1023/A:1018947420290.

FLOURIS, G.; BIKAKIS, A. A comprehensive study of argumentation frameworks with sets of attacking arguments. **International Journal of Approximate Reasoning**, [Amsterdam], v. 109, p. 55–86, 2019. DOI: 10.1016/j.ijar.2019.03.006.

GABBAY, D. M. Logical foundations for bipolar and tripolar argumentation networks: preliminary results. **Journal of Logic and Computation**, [Oxford], v. 26, n. 1, p. 247–292, 2016. DOI: 10.1093/logcom/ext027.

GOROGIANNIS, N.; HUNTER, A. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. **Artificial Intelligence**, [Amsterdam], v. 175, n. 9-10, p. 1479–1497, 2011. DOI: 10.1016/j.artint.2010.12.003.

GOTTIFREDI, S. *et al.* Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. **Artificial Intelligence**, [Amsterdam], v. 262, p. 336–368, 2018. DOI: 10.1016/j.artint.2018.06.008.

KÖNIG, M.; RAPBERGER, A.; ULBRICHT, M. Just a matter of perspective. *In:* INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELS OF ARGUMENT, 9., Cardiff, Wales. **Proceedings** […]. Amsterdam: IOS Press, 2022, p. 212–223. DOI: 10.3233/FAIA220154.

LAGASQUIE-SCHIEX, M.-C. Handling support cycles and collective interactions in the logical encoding of higher-order bipolar argumentation frameworks. **Journal of Logic and Computation**, [Oxford], v. 33, n. 2, p. 289–318, 2023. DOI: 10.1093/logcom/exac095.

MARTIN, C. *et al.* Attack semantics and collective attacks revisited. **Argument & Computation**, [Amsterdam], v. 16, n. 2, p. 151–211, 2025. DOI: 10.3233/AAC-230011.

NIELSEN, S. H.; PARSONS, S. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. *In:* INTERNATIONAL WORKSHOP ON ARGUMENTATION IN MULTI-AGENT SYSTEMS, 3., 2006, Hakodate, Japan. **Proceedings** […]. Berlin, Germany: Springer, 2007, p. 54–73. DOI: 10.1007/978-3-540-75526-5_4.

NIEVES, J. C.; CORTÉS, U.; OSORIO, M. Preferred extensions as stable models. **Theory and Practice of Logic Programming**, [Cambridge], v. 8, n. 4, p. 527–543, 2008. DOI: 10.1017/S1471068408003359.

NOUIOUA, F.; BOUTOUHAMI, S. Argumentation frameworks with necessities and their relationship with logic programs. **Argument & Computation**, [Amsterdam], v. 14, n. 1, p. 17–58, 2023. DOI: 10.3233/AAC-210028.

NOUIOUA, F.; RISCH, V. Argumentation frameworks with necessities. *In:* INTERNATIONAL CONFERENCE ON SCALABLE UNCERTAINTY MANAGEMENT, 5., 2011, Dayton, Ohio. **Proceedings** […]. Berlin, Germany: Springer, 2011, p. 163–176.

NOUIOUA, F.; RISCH, V. Bipolar argumentation frameworks with specialized supports. *In:* INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 22., 2010, Arras, France. **Proceedings** […]. [Washington, DC]: IEEE Computer Society, 2010, p. 215–218. DOI: 10.1109/ICTAI.2010.37.

OREN, N.; NORMAN, T. J. Semantics for Evidence-Based Argumentation. *In:* INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELS OF ARGUMENT, 2., 2008, Toulouse, France. **Proceedings** […]. Amsterdam: IOS Press, 2008, p. 276–284.

OREN, N.; REED, C.; LUCK, M. Moving between argumentation frameworks. *In:* INTERNATIONAL CONFERENCE ON COMPUTATIONAL MODELS OF ARGUMENT, 3., 2010, Desenzano del Garda. **Proceedings** […]. Amsterdam: IOS Press, 2010, p. 379–390.

POLBERG, S. **Developing the abstract dialectical framework**. 2017. Ph.D. Thesis (Doctorate in Computer Science) – Faculty of Informatics, Technische Universität Wien, Vienna, Austria, 2017. DOI: 10.34726/hss.2017.47655.

POTYKA, N. Abstract argumentation with markov networks. *In:* EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 24., 2020, Santiago de Compostela. **Proceedings** […]. Amsterdam: IOS Press, 2020a, p. 865–872. DOI: 10.3233/FAIA200177.

POTYKA, N. Bipolar abstract argumentation with dual attacks and supports. *In:* INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 17., 2020, Rhodes, Greece (Online). **Proceedings** […]. [*S. l.*]: IJCAI, 2020b, p. 677–686. DOI: 10.24963/kr.2020/69.

POTYKA, N. Generalizing complete semantics to bipolar argumentation frameworks. *In:* EUROPEAN CONFERENCE ON SYMBOLIC AND QUANTITATIVE APPROACHES WITH UNCERTAINTY, 16., 2021, Prague, Czech Republic. **Proceedings** […]. Cham, Switzerland: Springer, 2021, p. 130–143. DOI: 10.1007/978-3-030-86772-0_10.

PRAKKEN, H. Relating abstract and structured accounts of argumentation dynamics: the case of expansions. *In:* PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON

PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 20., 2023, Rhodes, Greece. **Proceedings** [...]. [*S. l.*]: IJCAI, 2023, p. 562–571. DOI: 10.24963/kr.2023/55.

PRZYMUSINSKI, T. Well-founded semantics coincides with three-valued stable semantics. **Fundamenta Informaticae**, [Amsterdam], v. 13, n. 4, p. 445–463, 1990. DOI: 10.3233/FI-1990-13404.

PÜHRER, J. Realizability of three-valued semantics for abstract dialectical frameworks. **Artificial Intelligence**, [Amsterdam], v. 278, p. 103198, 2020. DOI: 10.1016/j.artint.2019.103198.

RAPBERGER, A. Defining Argumentation Semantics under a Claim-centric View. *In:* INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 17., 2020, Rhodes, Greece (Online). **Proceedings** [...]. [*S. l.*]: IJCAI, 2020, p. 341–350. DOI: 10.24963/kr.2020/35.

RAPBERGER, A. **Unpacking the argument: a claim-centric view on abstract argumentation**. 2023. Ph.D. Thesis (Doctorate in Computer Science) – Faculty of Informatics, Technische Universität Wien, Vienna, Austria, 2023. DOI: 10.34726/hss.2023.112445.

RAPBERGER, A.; ULBRICHT, M. On dynamics in structured argumentation formalisms. **Journal of Artificial Intelligence Research**, [El Segundo], v. 77, p. 563–643, 2023. DOI: 10.1613/jair.1.14481.

ROCHA, V. H. N.; COZMAN, F. G. Bipolar Argumentation Frameworks with Explicit Conclusions: Connecting Argumentation and Logic Programming. *In:* INTERNATIONAL WORKSHOP ON NON-MONOTONIC REASONING, 20., 2022, Haifa, Israel. **Proceedings** [...]. [Aachen]: CEUR-WS.org, 2022, p. 49–60.

SÁ, S.; ALCÂNTARA, J. An abstract argumentation and logic programming comparison based on 5-valued labellings. *In:* EUROPEAN CONFERENCE ON SYMBOLIC AND QUANTITATIVE APPROACHES WITH UNCERTAINTY, 16., 2021, Prague, Czech Republic. **Proceedings** [...]. Cham, Switzerland: Springer, 2021a, p. 159–172. DOI: 10.1007/978-3-030-86772-0_12.

SÁ, S.; ALCÂNTARA, J. Assumption-based argumentation is logic programming with projection. *In:* EUROPEAN CONFERENCE ON SYMBOLIC AND QUANTITATIVE APPROACHES WITH UNCERTAINTY, 16., 2021, Prague, Czech Republic. **Proceedings** [...]. Cham, Switzerland: Springer, 2021b, p. 173–186. DOI: 10.1007/978-3-030-86772-0_13.

STRASS, H. Approximating operators and semantics for abstract dialectical frameworks. **Artificial Intelligence**, [Amsterdam], v. 205, p. 39–70, 2013. DOI: 10.1016/j.artint.2013.09.004.

STRASS, H. Expressiveness of two-valued semantics for abstract dialectical frameworks. **Journal of Artificial Intelligence Research**, [El Segundo], v. 54, p. 193–231, 2015. DOI: 10.1613/jair.4879.

TARSKI, A. A lattice-theoretical fixpoint theorem and its applications. **Pacific Journal of Mathematics**, [Berkeley, California], v. 5, n. 2, p. 285–309, 1955. DOI: 10.2140/pjm.1955.5.285.

TONI, F.; SERGOT, M. Argumentation and answer set programming. *In:* BALDUCCINI, M.; SON, T. C. (eds.). **Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday**. Berlin, Germany: Springer, 2011. p. 164–180. DOI: 10.1007/978-3-642-20832-4_11.

WU, Y.; CAMINADA, M.; GABBAY, D. M. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. **Studia logica**, [Amsterdam], v. 93, p. 383, 2009. DOI: 10.1007/s11225-009-9210-5.

APPENDIX A – PROOFS

Theorems and proofs from Chapter 2.

Theorems and proofs from Section 2.1.

Theorem 2. Let $\mathcal{A}$ be a finite set of arguments. Let $\mathbf{H} \subseteq 2^{\mathcal{A}}$ be a family of subsets of $\mathcal{A}$ where there is no $\mathcal{H}, \mathcal{H}' \in \mathbf{H}$ such that $\mathcal{H} \subseteq \mathcal{H}'$. Then $\text{Tr}[\text{Tr}[\mathbf{H}]] = \mathbf{H}$.

Proof. Recall that any transversal of $\mathbf{H}$ according to Definition 3 (here, we know $\mathbf{H}$ is a hypergraph on $\mathcal{A}$) is also a transversal of $\mathbf{H}$ according to Definition 5 (any hypergraph on $\mathcal{A}$ is a family of subsets of $\mathcal{A}$). A transversal of $\mathbf{H}$ according to Definition 5 (here, we only know $\mathbf{H}$ is a family of subsets of $\mathcal{A}$) will not be a transversal of $\mathbf{H}$ according to Definition 3 only if $\mathbf{H}$ is not a hypergraph on $\mathcal{A}$, which happens when either $\emptyset \in \mathbf{H}$, or $\bigcup_{\mathcal{H} \in \mathbf{H}} \mathcal{H} \neq \mathcal{A}$.

The result we wish to prove follows trivially from Theorem 1 if we consider that $\mathbf{H}$ is a hypergraph on $\mathcal{A}$. Now we consider the scenarios in which $\mathbf{H}$ is not a hypergraph on $\mathcal{A}$:

a) assume $\emptyset \in \mathbf{H}$. By the minimality condition, $\mathbf{H} = \{\emptyset\}$ and there is no transversal of $\mathbf{H}$. This means $\text{Tr}[\mathbf{H}] = \emptyset$. In this case, $\text{Tr}[\text{Tr}[\mathbf{H}]] = \text{Tr}[\emptyset]$. Any set $\mathcal{T} \subseteq \mathcal{A}$ intersects each element of $\emptyset$ (vacuously). Thus, there exists only one $\subseteq$-minimal transversal of $\emptyset$: the empty set $\emptyset$. It follows $\text{Tr}[\emptyset] = \{\emptyset\}$ and thus $\text{Tr}[\text{Tr}[\mathbf{H}]] = \mathbf{H}$;

b) assume $\bigcup_{\mathcal{H} \in \mathbf{H}} \mathcal{H} \neq \mathcal{A}$ (and $\emptyset \notin \mathbf{H}$). Define $\mathcal{A}' = \bigcup_{\mathcal{H} \in \mathbf{H}} \mathcal{H}$. Although $\mathbf{H}$ is not a hypergraph on $\mathcal{A}$, it is a hypergraph on $\mathcal{A}'$. By the minimality condition, $\mathbf{H}$ is simple. By Theorem 1, $\text{Tr}(\text{Tr}(\mathbf{H})) = \mathbf{H}$. As $\text{Tr}(\text{Tr}(\mathbf{H}))$ is defined, it follows $\text{Tr}(\text{Tr}(\mathbf{H})) = \text{Tr}[\text{Tr}[\mathbf{H}]]$ and thus $\text{Tr}[\text{Tr}[\mathbf{H}]] = \mathbf{H}$.

□

Theorems and proofs from Chapter 3.

Theorems and proofs from Section 3.2.

Proposition 3. Let $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$ in *BAF* $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. The following holds:

a) $\mathcal{S}$ defeats A iff $\mathcal{S}$ defeats A' for every $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{S}$ defeats A'' for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$;

b) $\mathcal{S}$ defends A iff $\mathcal{S}$ defends A' for some $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Proof. We will prove the following results:

a) $\mathcal{S}$ defeats A iff $\mathcal{S}$ defeats A' for every $A' \in \mathfrak{Sup}(A)$.

Assume S defeats A. By definition, (i) for every $A' \in \mathfrak{Sup}(A)$ there exists $B \in$ *Att*(A') such that $B \in S$. Let $C \in \mathfrak{Sup}(A)$. Since $\mathfrak{Sup}(C) \subseteq \mathfrak{Sup}(A)$, then by (i) we conclude that for every $A' \in \mathfrak{Sup}(C)$ there exists $B \in$ *Att*(A') such that $B \in S$. Therefore, if S defeats A, then S defeats A' for every $A' \in \mathfrak{Sup}(A)$. The converse is trivial because A is a supporter of A;

b) $\mathcal{S}$ defeats A iff $\mathcal{S}$ defeats A'' for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Assume $\mathcal{S}$ defeats A'' for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$. From the previous result, $\mathcal{S}$ defeats A' for every $A' \in \mathfrak{Sup}(A'')$. In particular, $\mathcal{S}$ defeats A. The converse is trivial because A is a supporter of A;

c) $\mathcal{S}$ defends A iff $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Now assume S defends A. By definition, (ii) there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in$ *Att*(A') it holds S defeats B. Let $C \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(C)$. Since $\mathfrak{Sup}(A) \subseteq \mathfrak{Sup}(C)$, then by (ii) we conclude that there exists $A' \in \mathfrak{Sup}(C)$ such that for every $B \in$ *Att*(A') it holds S defeats B. Therefore, if S defends A, then $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$. The converse is trivial because A is a supporter of A;

d) $\mathcal{S}$ defends A iff $\mathcal{S}$ defends A' for some $A' \in \mathfrak{Sup}(A)$.

Assume $\mathcal{S}$ defends A' for some $A' \in \mathfrak{Sup}(A)$. From the previous result, $\mathcal{S}$ defends A'' for every $A'' \in \mathcal{A}$ such that $A' \in \mathfrak{Sup}(A'')$. In particular, $\mathcal{S}$ defends A. The converse is trivial because A is a supporter of A.

□

Theorem 4. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}^* = (\mathcal{A}, \mathit{Att}, \mathit{Sup}^*)$ be *BAF*s such that Sup^* is the reflexive and transitive closure of *Sup*, i.e., $\mathit{Sup}^* = \mathfrak{Sup}$. For any set $\mathcal{S} \subseteq \mathcal{A}$, the following holds:

a) $F_{\mathcal{B}}(\mathcal{S}) = F_{\mathcal{B}^*}(\mathcal{S})$;

b) $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff $\mathcal{S}$ is conflict-free in $\mathcal{B}^*$.

Proof. Denote $\mathfrak{Sup}_{\mathcal{B}}(A) = \{A' \in \mathcal{A} \mid A' \text{ is a supporter of } A \text{ in } \mathcal{B}\}$.

Initially, we show that $\mathfrak{Sup}_{\mathcal{B}}(A) = \mathfrak{Sup}_{\mathcal{B}^*}(A)$ for every $A \in \mathcal{A}$. In fact, $B \in \mathfrak{Sup}_{\mathcal{B}}(A)$ iff $(B, A) \in \mathit{Sup}^*$ iff B is a supporter of A in $\mathcal{B}^*$ iff $B \in \mathfrak{Sup}_{\mathcal{B}^*}(A)$.

Then, $A \in F_{\mathcal{B}}(\mathcal{S})$ iff $\mathcal{S}$ defends A in $\mathcal{B}$ iff $\exists A' \in \mathfrak{Sup}_{\mathcal{B}}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $\mathcal{S}$ defeats B in $\mathcal{B}$ iff $\exists A' \in \mathfrak{Sup}_{\mathcal{B}}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $\forall B' \in \mathfrak{Sup}_{\mathcal{B}}(B)$

there exists $C \in \mathit{Att}(B')$ such that $C \in \mathcal{S}$ iff $\exists A' \in \mathfrak{Sup}_{\mathcal{B}^*}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $\forall B' \in \mathfrak{Sup}_{\mathcal{B}^*}(B)$ there exists $C \in \mathit{Att}(B')$ such that $C \in \mathcal{S}$ iff $\exists A' \in \mathfrak{Sup}_{\mathcal{B}^*}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $\mathcal{S}$ defeats B in $\mathcal{B}^*$ iff $\mathcal{S}$ defends A in $\mathcal{B}^*$ iff $A \in F_{\mathcal{B}^*}(\mathcal{S})$. We conclude that $F_{\mathcal{B}}(\mathcal{S}) = F_{\mathcal{B}^*}(\mathcal{S})$.

Finally, $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff $\forall A \in \mathcal{S}$ it holds $\mathcal{S}$ does not defeat A in $\mathcal{B}$ iff $\forall A \in \mathcal{S}$ there exists $A' \in \mathfrak{Sup}_{\mathcal{B}}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $B \notin \mathcal{S}$ iff $\forall A \in \mathcal{S}$ there exists $A' \in \mathfrak{Sup}_{\mathcal{B}^*}(A)$ such that $\forall B \in \mathit{Att}(A')$ it holds $B \notin \mathcal{S}$ iff $\forall A \in \mathcal{S}$ it holds $\mathcal{S}$ does not defeat A in $\mathcal{B}^*$ iff $\mathcal{S}$ is conflict-free in $\mathcal{B}^*$. □

Corollary 5. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ and $\mathcal{B}^* = (\mathcal{A}, \mathit{Att}, \mathit{Sup}^*)$ be *BAF*s such that $\mathit{Sup}^* = \mathfrak{Sup}$. The following holds:

a) $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}^*$;
b) $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-complete extension of $\mathcal{B}^*$;
c) $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}^*$;
d) $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}^*$;
e) $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a a β-stable extension of $\mathcal{B}^*$;
f) $\mathcal{S}$ is a β-semi-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a a β-semi-stable extension of $\mathcal{B}^*$.

Proof. This is straightforward from Theorem 4. □

Proposition 6. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* such that $\mathit{Sup} = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, \mathit{Att})$ be the associated *AAF*. For $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$, the following holds:

a) $\mathcal{S}$ defeats A in $\mathcal{B}$ iff $\mathcal{S}$ defeats A in $\mathfrak{F}$;
b) $\mathcal{S}$ defends A in $\mathcal{B}$ iff $\mathcal{S}$ defends A in $\mathfrak{F}$;
c) $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff $\mathcal{S}$ is conflict-free in $\mathfrak{F}$.

Proof. Assume $\mathit{Sup} = \emptyset$. Let $\mathcal{S} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$. Note that $\mathfrak{Sup}(A) = \{A\}$, i.e., A is the only supporter of A. Then, the following holds:

a) $\mathcal{S}$ defeats A in $\mathcal{B}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in \mathit{Att}(A')$ such that $B \in \mathcal{S}$ iff there exists an argument $B \in \mathit{Att}(A)$ such that $B \in \mathcal{S}$;
b) $\mathcal{S}$ defends A in $\mathcal{B}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in \mathit{Att}(A')$, it holds $\mathcal{S}$ defeats B in $\mathcal{B}$ iff for all $B \in \mathit{Att}(A)$ it holds $\mathcal{S}$ defeats B in $\mathcal{B}$ iff (by the previous item) for all $B \in \mathit{Att}(A)$, there exists $B' \in \mathit{Att}(B)$ such that $B' \in \mathcal{S}$ iff $\mathcal{S}$ defends A in $\mathfrak{A}$;

c) $\mathcal{S}$ is conflict-free in $\mathcal{B}$ iff for every $A \in \mathcal{S}$, it holds $\mathcal{S}$ does not defeat A in $\mathcal{B}$ iff for any $A \in \mathcal{S}$, there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in \mathit{Att}(A')$, it holds $B \notin \mathcal{S}$ iff for any $A, B \in \mathcal{S}$ it holds $(B, A) \notin \mathit{Att}$ iff $\mathcal{S}$ is conflict-free in $\mathfrak{A}$.

□

Corollary 7. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* such that $\mathit{Sup} = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, \mathit{Att})$ be the corresponding *AAF*. For $\mathcal{S} \subseteq \mathcal{A}$, the following holds:

a) $\mathcal{S}$ is a β-admissible extension of $\mathcal{B}$ iff $\mathcal{S}$ is an admissible extension of $\mathfrak{F}$;
b) $\mathcal{S}$ is a β-complete extension of $\mathcal{B}$ iff $\mathcal{S}$ is a complete extension of $\mathfrak{F}$;
c) $\mathcal{S}$ is a β-grounded extension of $\mathcal{B}$ iff $\mathcal{S}$ is a grounded extension of $\mathfrak{F}$;
d) $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a preferred extension of $\mathfrak{F}$;
e) $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a stable extension of $\mathfrak{F}$;
f) $\mathcal{S}$ is a β-semi-stable extension of $\mathcal{B}$ iff $\mathcal{S}$ is a semi-stable extension of $\mathfrak{F}$.

Proof. It follows straightforward from Proposition 6, and Definitions 12, 27 and 28. □

Lemma 8 (Fundamental Lemma)**.** Let $\mathcal{B}$ be a *BAF*, $\mathcal{S}$ be a β-admissible set of arguments, and A, A' be arguments which are acceptable with respect to $\mathcal{S}$. Then, the following holds:

a) $\mathcal{S}' = \mathcal{S} \cup \{A\}$ is β-admissible;
b) A' is acceptable with respect to $\mathcal{S}'$.

Proof. We prove each item:

a) it suffices to show $\mathcal{S}'$ is context-free. By absurd, assume S' is not conflict-free. This means there exists an argument $B \in \mathcal{S}'$ such that $\mathcal{S}'$ defeats B, i.e., for every $B' \in \mathfrak{Sup}(B)$, there exists an argument $C \in \mathit{Att}(B')$ such that $C \in \mathcal{S}$.

There are two possibilities:

– $B \in \mathcal{S}$: in this case, as $\mathcal{S}$ defends B, there exists $B' \in \mathfrak{Sup}(B)$ such that for all $C \in \mathit{Att}(B')$, it holds $\mathcal{S}$ defeats C in $\mathcal{B}$. As $\mathcal{S}'$ defeats B, it follows there exists an argument $C' \in \mathit{Att}(B')$ such that $C' \in \mathcal{S}'$. As $\mathcal{S}$ defeats C', we obtain $C' \notin \mathcal{S}$ because $\mathcal{S}$ is conflict-free. Thus C' is the argument A and $\mathcal{S}$ defeats A. As $\mathcal{S}$ also defends A, it is an absurd given that $\mathcal{S}$ is conflict-free;

– B is the argument A: given that $\mathcal{S}$ defends A and S' defeats A, we know there exists $A' \in \mathfrak{Sup}(A)$ and there exists $C \in \mathit{Att}(A')$ such that $C \in \mathcal{S}'$ and $\mathcal{S}$ defeats C in $\mathcal{B}$. Argument $C \notin \mathcal{S}$ as $\mathcal{S}$ is conflict-free. Thus C is the argument A and $\mathcal{S}$ defeats A. As $\mathcal{S}$ also defends A, it is an absurd given that $\mathcal{S}$ is conflict-free;

b) As $A' \in F(\mathcal{S})$, it is clear $A' \in F(\mathcal{S} \cup \{A\})$, i.e., A' is acceptable with respect to $\mathcal{S}'$.

□

Theorem 9. Let $\mathcal{B}$ be a *BAF*. The following holds:

a) the set of all β-admissible sets of $\mathcal{B}$ forms a complete partial order with respect to set inclusion;

b) for each β-admissible extension $\mathcal{S}$ of $\mathcal{B}$, there exists a maximal β-admissible extension $\mathcal{E}$ of $\mathcal{B}$ such that $\mathcal{S} \subseteq \mathcal{E}$.

Proof. Let $ADM_{\mathcal{B}} = \{\mathcal{S} \mid \mathcal{S} \textit{ is a } \beta\textit{-admissible extension of } \mathcal{B}\}$. We will show $(ADM_{\mathcal{B}}, \subseteq)$ is a complete partially ordered set:

a) Let $(\mathfrak{D}, \subseteq)$ be a directed set with $\mathfrak{D} \subseteq ADM_{\mathcal{B}}$. We have to prove

$$lub(\mathfrak{D}) = \bigcup\{\mathcal{S} \mid \mathcal{S} \in \mathfrak{D}\} \in ADM_{\mathcal{B}}. \tag{A.1}$$

We proceed in parts:

– $lub(\mathfrak{D})$ is conflict-free. By absurd, suppose $lub(\mathfrak{D})$ is not conflict-free. This means there exists argument $A \in lub(\mathfrak{D})$ such that $lub(\mathfrak{D})$ defeats A. As $A \in lub(\mathfrak{D})$, there exists $\mathcal{S} \in \mathfrak{D}$ such that $\mathcal{S}$ defends A. Hence, there exists $A' \in \mathfrak{Sup}(A)$ and $B \in Att(A') \cap lub(\mathfrak{D})$ such that $\mathcal{S}$ defeats B in $\mathcal{B}$. As $B \in lub(\mathfrak{D})$, there exists $\mathcal{S}' \in \mathfrak{D}$ such that $\mathcal{S}'$ defends B. Given that $(\mathfrak{D}, \subseteq)$ is a directed set, there exists $\mathcal{S}'' \in \mathfrak{D}$ such that $\mathcal{S} \subseteq S''$ and $\mathcal{S}' \subseteq S''$. It follows that $\mathcal{S}''$ both defeats B and defends B. It is an absurd as S'' is conflict-free;

– $lub(\mathfrak{D}) \subseteq F_{\mathcal{B}}(lub(\mathfrak{D}))$: if $A \in lub(\mathfrak{D}) \Rightarrow$ there exists $\mathcal{S} \in \mathfrak{D} \subseteq ADM_{\mathcal{B}}$ such that $A \in \mathcal{S} \Rightarrow$ there exists $\mathcal{S} \in \mathfrak{D}$ such that $A \in F_{\mathcal{B}}(\mathcal{S}) \Rightarrow$ in consequence of Lemma 8, $A \in F_{\mathcal{B}}(lub(\mathfrak{D})) = F_{\mathcal{B}}(\bigcup\{\mathcal{S} \mid \mathcal{S} \in \mathfrak{D}\})$;

b) Let $\mathfrak{S} \subseteq \{\mathcal{S}' \mid \mathcal{S}' \textit{ is a } \beta\textit{-admissible extension of } \mathcal{B} \textit{ and } \mathcal{S} \subseteq \mathcal{S}'\}$ such that $(\mathfrak{S}, \subseteq)$ is a directed set. According to the previous item, $\mathcal{E} = lub(\mathfrak{S})$ is a β-admissible extension of $\mathcal{B}$. Indeed, by definition, $\mathcal{E}$ is a $\subseteq$-maximal β-admissible extension of $\mathcal{B}$ and $\mathcal{S} \subseteq \mathcal{E}$.

□

Proposition 11. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. Then $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$ iff $\mathcal{S} = \{A \mid A \textit{ is not defeated by } \mathcal{S}\}$.

Proof. $\mathcal{S} = \{A \mid A \textit{ is not defeated by } \mathcal{S}\}$ iff $\mathcal{S} = \{A \mid \exists A' \in \mathfrak{Sup}(A)$ *such that* $\forall B \in$ *Att*(A'), *it holds* $B \notin \mathcal{S}\}$ and $\mathcal{S}$ is conflict-free and $\mathcal{S} \cup \mathcal{S}^+ = \mathcal{A}$ iff $\mathcal{S} = \{A \mid \exists A' \in \mathfrak{Sup}(A)$ *such that* $\forall B \in$ *Att*(A'), *it holds* B *is defeated by* $S\}$ and $\mathcal{S}$ is conflict-free and $\mathcal{S} \cup \mathcal{S}^+ = \mathcal{A}$ iff $\mathcal{S} = F_{\mathcal{B}}(\mathcal{S})$ and $\mathcal{S}$ is conflict-free and $\mathcal{S} \cup \mathcal{S}^+ = \mathcal{A}$ iff $\mathcal{S}$ is a β-stable extension of $\mathcal{B}$. □

Proposition 12. For any *BAF* $\mathcal{B}$, any β-stable extension of $\mathcal{B}$ is a β-preferred extension of $\mathcal{B}$. However, it is not always the case that every β-preferred extension of $\mathcal{B}$ is a β-stable extension of $\mathcal{B}$.

Proof. Let $\mathcal{S}$ be a β-stable extension of $\mathcal{B}$. By absurd, suppose $\mathcal{S}$ is not a β-preferred extension of $\mathcal{B}$. This means there exists a β-complete extension $\mathcal{S}'$ of $\mathcal{B}$ such that $\mathcal{S} \subset \mathcal{S}'$. As any argument in $\mathcal{S}' - \mathcal{S}$ is defeated by $\mathcal{S}$, if follows $\mathcal{S}'$ is not conflict-free. An absurd as S' is a complete extension of $\mathcal{B}$.

In order to show the reverse does not hold, we present the *BAF* $\mathcal{B} = (\mathcal{A}, \textit{Att}, \textit{Sup})$ with $\mathcal{A} = \{A\}$, $\textit{Att} = \{(A, A)\}$ and $\textit{Sup} = \emptyset$. It is clear that the empty set is a β-preferred extension of $\mathcal{B}$ which is clearly not stable. □

Lemma 13. Let $\mathcal{B}$ be a *BAF*. If $\mathcal{S}$ is a conflict-free set in $\mathcal{B}$, then $F_{\mathcal{B}}(\mathcal{S})$ is also conflict-free in $\mathcal{B}$.

Proof. Let $\mathcal{S}$ be a conflict-free set in $\mathcal{B}$. By absurd, assume $F_{\mathcal{B}}(\mathcal{S})$ is not conflict-free in $\mathcal{B}$. This means there exists an argument $A \in F_{\mathcal{B}}(\mathcal{S})$ such that $F_{\mathcal{B}}(\mathcal{S})$ defeats A, i.e., for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in \textit{Att}(A')$ such that $B \in F_{\mathcal{B}}(\mathcal{S})$. Then, as $\mathcal{S}$ defends A, we know there exists $B \in F_{\mathcal{B}}(\mathcal{S})$ such that $\mathcal{S}$ defeats B. But $\mathcal{S}$ also defends B. It is an absurd as $\mathcal{S}$ is conflict-free. □

Lemma 14. Let $\mathcal{B} = (\mathcal{A}, \textit{Att}, \textit{Sup})$ be a *BAF*. Then $F_{\mathcal{B}}(\mathcal{S})$ is monotonic with respect to set inclusion.

Proof. We have to prove that if $\mathcal{S} \subseteq \mathcal{S}'$, then $F_{\mathcal{B}}(\mathcal{S}) \subseteq F_{\mathcal{B}}(\mathcal{S}')$. This result is straightforward as if an argument A is defended by $\mathcal{S}$, then A is also defended by any superset of $\mathcal{S}$. □

Lemma 15. Let $\mathcal{B}$ be a *BAF*. It holds $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$ iff $\mathcal{S}$ is a $\subseteq$-maximal β-admissible extension of $\mathcal{B}$.

Proof. We prove each item:

a) let $\mathcal{S}$ be a $\subseteq$-maximal β-admissible extension (with respect to set inclusion) of $\mathcal{B}$, i.e., $\mathcal{S} \subseteq F_{\mathcal{B}}(\mathcal{S})$. From Lemma 13, it is clear $F_{\mathcal{B}}(\mathcal{S})$ is also conflict-free. From the monotonicity of $F_{\mathcal{B}}$ (Lemma 14), we obtain

$$F_{\mathcal{B}}(\mathcal{S}) \subseteq F_{\mathcal{B}}(F_{\mathcal{B}}(\mathcal{S})).$$

Hence, $F_{\mathcal{B}}(\mathcal{S})$ is also a β-admissible extension of $\mathcal{B}$. As $\mathcal{S} \subseteq F_{\mathcal{B}}(\mathcal{S})$ and $\mathcal{S}$ is a $\subseteq$-maximal β-admissible extension of $\mathcal{B}$, it follows $\mathcal{S} = F_{\mathcal{B}}(\mathcal{S})$ and consequently $\mathcal{S}$ is a β-preferred extension of $\mathcal{B}$;

b) let $\mathcal{S}$ be a β-preferred extension of $\mathcal{B}$. By absurd, assume $\mathcal{S}$ is not a $\subseteq$-maximal β-admissible extension of $\mathcal{B}$. Then there exists a $\subseteq$-maximal β-admissible extension $\mathcal{S}'$ of $\mathcal{B}$ such that $\mathcal{S} \subset \mathcal{S}'$. From the previous case, we obtain $\mathcal{S}'$ is also a β-preferred extension of $\mathcal{B}$. It is an absurd as $\mathcal{S} \subset \mathcal{S}'$.

□

Theorem 17. Every *BAF* $\mathcal{B}$ possesses a unique β-grounded extension and it is the $\subseteq$-least fixpoint of $F_{\mathcal{B}}$.

Proof. From Theorem 16, we know $\mathcal{B}$ has at least one β-complete extension $\mathcal{S}'$. From the monotonicity of $F_{\mathcal{B}}$ (Lemma 14) and the well-known Knaster-Tarski theorem (Tarski, 1955), we obtain the least fixpoint $\mathcal{S}$ of $F_{\mathcal{B}}$ exists. As $\mathcal{S} \subseteq \mathcal{S}'$, and $\mathcal{S}'$ is conflict-free, then $\mathcal{S}$ is also conflict-free. This means $\mathcal{S}$ is the unique β-grounded extension of $\mathcal{B}$. □

Theorem 18. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. The β-complete extensions of $\mathcal{B}$ form a complete semilattice with respect to set inclusion.

Proof. Let $ADM_{\mathcal{B}} = \{\mathcal{S} \mid \mathcal{S}$ *is a* β*-admissible extension of* $\mathcal{B}\}$, $COMP_{\mathcal{B}} = \{\mathcal{S} \mid \mathcal{S}$ *is a* β*-complete extension of* $\mathcal{B}\}$, and $\mathcal{G} \in COMP_{\mathcal{B}}$ the grounded extension of $\mathcal{B}$. We will show $(COMP_{\mathcal{B}}, \subseteq)$ is a complete semilattice:

a) first, we prove each nonempty subset of $COMP_{\mathcal{B}}$ has a greatest lower bound. Let $\mathfrak{S} \subseteq COMP_{\mathcal{B}}$ and $\mathfrak{S} \neq \emptyset$. We define

$$LB = \{\mathcal{E} \in ADM_{\mathcal{B}} \mid (\forall \mathcal{C}' \in \mathfrak{C})\mathcal{E} \subseteq \mathcal{C}'\}.$$

It is clear $\mathcal{G} \in LB$. So LB is not empty. Given that $\mathcal{E} \subseteq \mathcal{C}'$ for any $\mathcal{E} \in LB$ and for any $\mathcal{C}' \in \mathfrak{C}$, we obtain from Lemma 14, $F_{\mathcal{B}}(\mathcal{E}) \subseteq F_{\mathcal{B}}(\mathcal{C}') = \mathcal{C}'$, i.e.,

$$(\forall \mathcal{E} \in LB)(\forall \mathcal{C}' \in \mathfrak{C})[F_{\mathcal{B}}(\mathcal{E}) \subseteq \mathcal{C}'].$$

Thus

$$(\forall \mathcal{E} \in \mathit{LB})[F_{\mathcal{B}}(\mathcal{E}) \in \mathit{LB}]. \tag{A.2}$$

Let

$$\mathcal{C} = \bigcup \{\mathcal{E} \mid \mathcal{E} \in \mathit{LB}\}.$$

Now we have to prove

$$\mathcal{C} \in \mathit{ADM}_{\mathcal{B}}. \tag{A.3}$$

We show that $\mathcal{C}$ is conflict-free and defends each of its elements:

- $\mathcal{C}$ is conflict-free: by absurd, suppose $\mathcal{C}$ is not conflict-free. This means there exists argument $A \in \mathcal{C}$ such that $\mathcal{C}$ defeats A. As $A \in \mathcal{C}$, there exists $\mathcal{E} \in \mathit{LB}$ such that $\mathcal{E}$ defends A. Hence, there exists $A' \in \mathfrak{Sup}(A)$ and there exists $B \in \mathit{Att}(A') \cap \mathcal{C}$ such that $\mathcal{E}$ defeats B. As $B \in \mathcal{C}$, there exists $\mathcal{E}' \in \mathit{LB}$ such that $\mathcal{E}'$ defends B. It is clear $\mathcal{E} \cup \mathcal{E}' \in \mathit{LB}$ as both $\mathcal{E}$ and $\mathcal{E}'$ are in LB. It follows $\mathcal{E} \cup \mathcal{E}'$ both defeats B and defends B. It is an absurd as $\mathcal{E} \cup \mathcal{E}'$ is conflict-free;
- $\mathcal{C} \subseteq F_{\mathcal{B}}(\mathcal{C})$: if $A \in \mathcal{C} \Rightarrow$ there exists $\mathcal{E} \in \mathit{LB} \subseteq \mathit{ADM}_{\mathcal{B}}$ such that $A \in \mathcal{E} \Rightarrow$ there exists $\mathcal{E} \in \mathit{LB}$ such that $A \in F_{\mathcal{B}}(\mathcal{E}) \Rightarrow$ in consequence of Lemma 8, $A \in F_{\mathcal{B}}(\mathcal{C})$.

It follows from the definition of $\mathcal{C}$ that $(\forall \mathcal{C}' \in \mathfrak{C})[\mathcal{C} \subseteq \mathcal{C}']$. As a consequence, $\mathcal{C} \in \mathit{LB}$ and according to Equation (A.2), $F_{\mathcal{B}}(\mathcal{C}) \in \mathit{LB}$.

As $\mathcal{C} \in \mathit{ADM}_{\mathcal{B}}$ (Equation (A.3)), we know $\mathcal{C} \subseteq F_{\mathcal{B}}(\mathcal{C})$; as $F_{\mathcal{B}}(\mathcal{C}) \in \mathit{LB}$, we know $F_{\mathcal{B}}(\mathcal{C}) \subseteq \mathcal{C}$. Then $F_{\mathcal{B}}(\mathcal{C}) = \mathcal{C}$, i.e., $\mathcal{C} \in \mathit{COMP}_{\mathcal{B}}$.

The complete extension $\mathcal{C}$ is a lower bound of $\mathfrak{C}$ because for every $\mathcal{C}' \in \mathfrak{C}$, it holds $\mathcal{C} \subseteq \mathcal{C}'$. Furthermore, $\mathcal{C}$ is the greatest lower bound of $\mathfrak{C}$ as $\mathcal{C} \in \mathit{LB}$ and for every $\mathcal{E} \in \mathit{LB}$, it holds $\mathcal{E} \subseteq \mathcal{C}$;

b) For each chain $(\mathfrak{S}, \subseteq)$ of $(\mathit{COMP}_{\mathcal{B}}, \subseteq)$, the set $\mathfrak{S}$ has an least upper bound:
Firstly note $(\mathfrak{S}, \subseteq)$ is also a chain of $(\mathit{ADM}_{\mathcal{B}}, \subseteq)$, where $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$. This means $(\mathfrak{S}, \subseteq)$ is a directed set of $(\mathit{ADM}_{\mathcal{B}}, \subseteq)$. Then, according to Theorem 9

$$\mathcal{S} = \bigcup \{\mathcal{S}' \mid \mathcal{S}' \in \mathfrak{S}\} \in \mathit{ADM}_{\mathcal{B}}.$$

By Theorem 9, there exists a $\subseteq$-maximal β-admissible extension $\mathcal{E}$ of $\mathcal{B}$ such that $\mathcal{S} \subseteq \mathcal{E}$. We know $\mathcal{E} \subseteq F_{\mathcal{B}}(\mathcal{E})$. From the monotonicity of $F_{\mathcal{B}}$ (Lemma 14), we obtain $F_{\mathcal{B}}(\mathcal{E}) \subseteq F_{\mathcal{B}}(F_{\mathcal{B}}(\mathcal{E}))$ and from Lemma 13, we obtain $F_{\mathcal{B}}(\mathcal{E})$ is conflict-free. This means $F_{\mathcal{B}}(\mathcal{E}) \in ADM_{\mathcal{B}}$. As $\mathcal{E}$ is a $\subseteq$-maximal β-admissible extension $\mathcal{B}$, it follows $F_{\mathcal{B}}(\mathcal{E}) = \mathcal{E}$, i.e., $\mathcal{E} \in COMP_{\mathcal{B}}$; besides, $\mathcal{E}$ is an upper bound of $\mathfrak{C}$.

Let $\mathfrak{C}' = \{\mathcal{E} \mid \mathcal{E} \in COMP_{\mathcal{B}} \textit{ and } \mathcal{E} \textit{ is an upper bound of } \mathfrak{C}\}$. Obviously $\mathfrak{C}' \neq \emptyset$. Note also $\mathcal{S}$ is a lower bound of $\mathfrak{C}'$. From the previous item, we know $\mathfrak{C}'$ has a greatest lower bound $\mathcal{C}''$ in $(COMP_{\mathcal{B}}, \preceq)$.

As $\mathcal{S} \subseteq \mathcal{C}''$ we conclude $\mathcal{C}''$ is the least upper bound of $\mathfrak{C}$.

□

Theorems and Proofs from Section 3.3.

Proposition 19. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathcal{L}$ be a β-complete labeling of $\mathcal{B}$. For any $A \in \mathcal{A}$, the following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A'') = \texttt{in}$ for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$;

b) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A'') = \texttt{out}$ for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Proof. We will prove the following results:

a) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A)$.

Assume $\mathcal{L}(A) = \texttt{out}$. By definition, (i) for every $A' \in \mathfrak{Sup}(A)$ there exists $B \in Att(A')$ such that $\mathcal{L}(B) = \texttt{in}$. Let $C \in \mathfrak{Sup}(A)$. Since $\mathfrak{Sup}(C) \subseteq \mathfrak{Sup}(A)$, then by (i) we conclude that for every $A' \in \mathfrak{Sup}(C)$ there exists $B \in Att(A')$ such that $\mathcal{L}(B) = \texttt{in}$, i.e., $\mathcal{L}(C) = \texttt{out}$. Therefore, if $\mathcal{L}(A) = \texttt{out}$, then $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A)$. The converse is trivial, because A is a supporter of A;

b) $\mathcal{L}(A) = \texttt{out}$ iff $\mathcal{L}(A'') = \texttt{out}$ for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Assume $\mathcal{L}(A'') = \texttt{out}$ for some $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$. From the previous result, $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A'')$. In particular, $\mathcal{L}(A) = \texttt{out}$. The converse is trivial, because A is a supporter of A;

c) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(A'') = \texttt{in}$ for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$.

Now assume $\mathcal{L}(A) = \texttt{in}$. By definition, (ii) there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$ it holds $\mathcal{L}(B) = \texttt{out}$. Let $C \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(C)$. Since $\mathfrak{Sup}(A) \subseteq \mathfrak{Sup}(C)$, then by (ii) we conclude that there exists $A' \in \mathfrak{Sup}(C)$ such that for every $B \in Att(A')$ it holds $\mathcal{L}(B) = \texttt{out}$, i.e., $\mathcal{L}(C) = \texttt{in}$. Therefore, if $\mathcal{L}(A) = \texttt{in}$, then $\mathcal{L}(A'') = \texttt{in}$ for every $A'' \in \mathcal{A}$ such that $A \in \mathfrak{Sup}(A'')$. The converse is trivial, because A is a supporter of A;

d) $\mathcal{L}(A) = \texttt{in}$ iff $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(A)$.
Assume $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(A)$. From the previous result, $\mathcal{L}(A'') = \texttt{in}$ for every $A'' \in \mathcal{A}$ such that $A' \in \mathfrak{Sup}(A'')$. In particular, $\mathcal{L}(A) = \texttt{in}$. The converse is trivial, because A is a supporter of A.

□

Proposition 20. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathcal{L}$ a labeling of $\mathcal{B}$. Then, $\mathcal{L}$ is β-complete iff for any $A \in \mathcal{A}$, all conditions below are satisfied:

a) $\mathcal{L}(A) = \texttt{in}$ iff $\texttt{in}(\mathcal{L})$ defends A;
b) $\mathcal{L}(A) = \texttt{out}$ iff $\texttt{in}(\mathcal{L})$ defeats A.

Proof.

($\Longrightarrow$) Let $\mathcal{L}$ be a β-complete labeling. Firstly we will prove

$$\mathcal{L}(A) = \texttt{out} \textit{ iff } \texttt{in}(\mathcal{L}) \textit{ defeats } A. \tag{A.4}$$

It holds $\mathcal{L}(A) = \texttt{out}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in \mathit{Att}(A')$ such that $\mathcal{L}(B) = \texttt{in}$ iff $\texttt{in}(\mathcal{L})$ defeats A.

Now we will prove

$$\mathcal{L}(A) = \texttt{in} \textit{ iff } \texttt{in}(\mathcal{L}) \textit{ defends } A. \tag{A.5}$$

It holds $\mathcal{L}(A) = \texttt{in}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) = \texttt{out}$ iff according to Equation (A.4), there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in \mathit{Att}(A')$, it holds $\texttt{in}(\mathcal{L})$ defeats B iff $\texttt{in}(\mathcal{L})$ defends A.

($\Longleftarrow$) Assume that for any $A \in \mathcal{A}$, both Equations (A.4) and (A.5) hold. Then, from Equation (A.4) and Definition 25, we can infer that for any $A \in \mathcal{A}$, $\mathcal{L}(A) = \texttt{out}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in \mathit{Att}(A')$ such that $B \in \texttt{in}(\mathcal{L})$.

Similarly, from Equation (A.5) and Definition 25, we can infer that for any $A \in \mathcal{A}$, $\mathcal{L}(A) = \texttt{in}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in \mathit{Att}(A')$, it holds $\texttt{in}(\mathcal{L})$ defeats B in $\mathcal{B}$. From Equation (A.4), we obtain that for any $A \in \mathcal{A}$, $\mathcal{L}(A) = \texttt{in}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) = \texttt{out}$.

Hence, $\mathcal{L}$ is a β-complete labeling. □

Proposition 21. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* such that $Sup = \emptyset$, and $\mathfrak{F} = (\mathcal{A}, Att)$ the corresponding *AAF*. For a labeling $\mathcal{L}$ of $\mathcal{B}$, we can guarantee the following properties:

a) $\mathcal{L}$ is a β-admissible labeling of $\mathcal{B}$ iff $\mathcal{L}$ is an admissible labeling of $\mathfrak{F}$;

b) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{F}$;

c) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a grounded labeling of $\mathfrak{F}$;

d) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a preferred labeling of $\mathfrak{F}$;

e) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a stable labeling of $\mathfrak{F}$;

f) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{F}$.

Proof. These results hold because Definition 29 (β-admissible labelings) and Definition 28 (labeling-based semantics for *BAF*s) collapse into the definition of *AAF* admissible labelings and Definition 14 (labeling-based semantics for *AAF*s) respectively when $Sup = \emptyset$. □

Lemma 27. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$;

b) $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2)$.

Proof. We start by proving $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$:

a) ($\Longrightarrow$) suppose $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$. It follows $A \in \texttt{out}(\mathcal{L}_1) \Rightarrow$ for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in Att(A')$ such that $B \in \texttt{in}(\mathcal{L}_1)$. $\Rightarrow$ (as $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$) for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in Att(A')$ such that $B \in \texttt{in}(\mathcal{L}_2) \Rightarrow A \in \texttt{out}(\mathcal{L}_2)$;

b) ($\Longleftarrow$) suppose $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$. It follows $A \in \texttt{in}(\mathcal{L}_1) \Rightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $B \in \texttt{out}(\mathcal{L}_1) \Rightarrow$ (as $\texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$) there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $B \in \texttt{out}(\mathcal{L}_2) \Rightarrow A \in \texttt{in}(\mathcal{L}_2)$.

We now prove $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$ iff $\texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2)$:

a) ($\Longrightarrow$) it follows $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2) \Rightarrow \texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and $\texttt{in}(\mathcal{L}_2) \nsubseteq \texttt{in}(\mathcal{L}_1) \Rightarrow \texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_2) \nsubseteq \texttt{out}(\mathcal{L}_1) \Rightarrow \texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2)$;

b) ($\Longleftarrow$) it follows $\texttt{out}(\mathcal{L}_1) \subset \texttt{out}(\mathcal{L}_2) \Rightarrow \texttt{out}(\mathcal{L}_1) \subseteq \texttt{out}(\mathcal{L}_2)$ and $\texttt{out}(\mathcal{L}_2) \nsubseteq \texttt{out}(\mathcal{L}_1) \Rightarrow \texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ and $\texttt{in}(\mathcal{L}_2) \nsubseteq \texttt{in}(\mathcal{L}_1) \Rightarrow \texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$.

□

Lemma 28. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) If $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subseteq \texttt{undec}(\mathcal{L}_1)$;

b) If $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$, then $\texttt{undec}(\mathcal{L}_2) \subset \texttt{undec}(\mathcal{L}_1)$;

c) if $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subseteq \mathtt{undec}(\mathcal{L}_1)$;

d) if $\mathtt{out}(\mathcal{L}_1) \subset \mathtt{out}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subset \mathtt{undec}(\mathcal{L}_1)$.

Proof. We prove each item:

a) proof of if $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subseteq \mathtt{undec}(\mathcal{L}_1)$.

   Suppose $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$. From Lemma 27, it follows $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$. Then $A \in \mathtt{undec}(\mathcal{L}_2) \Rightarrow A \notin \mathtt{in}(\mathcal{L}_2) \cup \mathtt{out}(\mathcal{L}_2) \Rightarrow A \notin \mathtt{in}(\mathcal{L}_1) \cup \mathtt{out}(\mathcal{L}_1) \Rightarrow A \in \mathtt{undec}(\mathcal{L}_1)$;

b) proof of if $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subset \mathtt{undec}(\mathcal{L}_1)$.

   Suppose $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$. From Lemma 27, it follows $\mathtt{out}(\mathcal{L}_1) \subset \mathtt{out}(\mathcal{L}_2)$. It also follows $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ and $\mathtt{in}(\mathcal{L}_2) \not\subseteq \mathtt{in}(\mathcal{L}_1)$ and $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$ and $\mathtt{out}(\mathcal{L}_2) \not\subseteq \mathtt{out}(\mathcal{L}_1)$.

   From the previous item, we obtain $\mathtt{undec}(\mathcal{L}_2) \subseteq \mathtt{undec}(\mathcal{L}_1)$. As $\mathtt{in}(\mathcal{L}_2) \not\subseteq \mathtt{in}(\mathcal{L}_1)$, there exists an argument such that $A \in \mathtt{in}(\mathcal{L}_2)$, but $A \notin \mathtt{in}(\mathcal{L}_1)$. It is clear $A \notin \mathtt{out}(\mathcal{L}_1)$ (otherwise, A would be in $\mathtt{out}(\mathcal{L}_2)$). Thus, $A \in \mathtt{undec}(\mathcal{L}_1)$. It implies $\mathtt{undec}(\mathcal{L}_1) \not\subseteq \mathtt{undec}(\mathcal{L}_2)$. Consequently, $\mathtt{undec}(\mathcal{L}_2) \subset \mathtt{undec}(\mathcal{L}_1)$;

c) proof of if $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subseteq \mathtt{undec}(\mathcal{L}_1)$.

   It is similar to the proof of item a);

d) proof of if $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$, then $\mathtt{undec}(\mathcal{L}_2) \subset \mathtt{undec}(\mathcal{L}_1)$.

   It is similar to the proof of item b).

□

Lemma 29. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of a *BAF* $\mathcal{B}$. The following holds:

a) if $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$, then $\mathcal{L}_1 = \mathcal{L}_2$;

b) if $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2)$, then $\mathcal{L}_1 = \mathcal{L}_2$.

Proof.

We prove each item:

a) it follows $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2) \Rightarrow \mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ and $\mathtt{in}(\mathcal{L}_2) \subseteq \mathtt{in}(\mathcal{L}_1) \Rightarrow$ from Lemma 27, $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$ and $\mathtt{out}(\mathcal{L}_2) \subseteq \mathtt{out}(\mathcal{L}_1) \Rightarrow \mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2) \Rightarrow$ as $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$, it holds $\mathtt{undec}(\mathcal{L}_1) = \mathtt{undec}(\mathcal{L}_2) \Rightarrow \mathcal{L}_1 = \mathcal{L}_2$ as $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$, $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2)$ and $\mathtt{undec}(\mathcal{L}_1) = \mathtt{undec}(\mathcal{L}_2)$;

b) it follows $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2) \Rightarrow \mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$ and $\mathtt{out}(\mathcal{L}_2) \subseteq \mathtt{out}(\mathcal{L}_1) \Rightarrow$ from Lemma 27, $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ and $\mathtt{in}(\mathcal{L}_2) \subseteq \mathtt{in}(\mathcal{L}_1) \Rightarrow \mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2) \Rightarrow$

as $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$, it holds $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2) \Rightarrow \mathcal{L}_1 = \mathcal{L}_2$ as $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$, $\texttt{out}(\mathcal{L}_1) = \texttt{out}(\mathcal{L}_2)$ and $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2)$.

□

Theorem 22. Let $\mathcal{B}$ be a *BAF*. Then the following holds:

a) if $\mathcal{L}$ is a β-admissible labeling of $\mathcal{B}$, then $\texttt{Lab2Ext}(\mathcal{L})$ is a β-admissible set of $\mathcal{B}$;

b) if $\mathcal{E}$ is a β-admissible set of $\mathcal{B}$, then $\texttt{Ext2Lab}(\mathcal{E})$ is a β-admissible labeling of $\mathcal{B}$.

Proof. We prove each item:

a) let $\mathcal{L}$ be a β-admissible labeling of $\mathcal{B}$. We have to guarantee $\mathcal{E} = \texttt{Lab2Ext}(\mathcal{L})$ is a β-admissible extension of $\mathcal{B}$:

   – by absurd, suppose $\mathcal{E}$ is not conflict-free. It follows there exists argument $A \in \mathcal{E}$ such that $A \in \mathcal{E}^+ = \texttt{in}(\mathcal{L})^+$.
   As $A \in \mathcal{E} = \texttt{in}(\mathcal{L})$, there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $\mathcal{L}(B) = \texttt{out}$.
   As $A \in \mathcal{E}^+ = \texttt{in}(\mathcal{L})^+$, it follows A is defeated by $\texttt{in}(\mathcal{L})$, i.e., for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in Att(A')$ such that $B \in \texttt{in}(\mathcal{L})$.
   It is an absurd, so $\mathcal{E}$ is conflict-free;

   – $A \in \mathcal{E} = \texttt{in}(\mathcal{L}), \Longrightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $\mathcal{L}(B) = \texttt{out} \Longrightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds for every $B' \in \mathfrak{Sup}(B)$, there exists $C \in Att(B')$ such that $\mathcal{L}(C) = \texttt{in}$ $\Longrightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in Att(A')$, it holds $\texttt{in}(\mathcal{L})$ defeats $B \Longrightarrow \texttt{in}(\mathcal{L})$ defends $A \Longrightarrow A \in F_{\mathcal{B}}(\texttt{in}(\mathcal{L}))$. Thus $\mathcal{E} \subseteq F_{\mathcal{B}}(\mathcal{E})$;

b) let $\mathcal{E}$ be a β-admissible extension of $\mathcal{B}$. We have to guarantee $\mathcal{L} = \texttt{Ext2Lab}(\mathcal{E})$ is a β-admissible labeling of $\mathcal{B}$:

   – $\mathcal{L}(A) = \texttt{in} \Longrightarrow A \in \mathcal{E} \Longrightarrow A \in F_{\mathcal{B}}(\mathcal{E}) \Longrightarrow A$ is defended by $\mathcal{E} \Longrightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in Att(A')$, it holds $\mathcal{E}$ defeats B in $\mathcal{B} \Longrightarrow$ there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in Att(A')$, it holds $B \in \texttt{out}(\mathcal{L})$;

   – $\mathcal{L}(A) = \texttt{out} \Longrightarrow A \in \mathcal{E}^+ \Longrightarrow A$ is defeated by $\mathcal{E} \Longrightarrow$ for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in Att(A')$ such that $B \in \mathcal{E} \Longrightarrow$ for every $A' \in \mathfrak{Sup}(A)$, there exists $B \in Att(A')$ such that $B \in \texttt{in}(\mathcal{L})$.

□

Theorem 23. When restricting their domains respectively to β-complete extensions and β-complete labelings, the functions $\texttt{Ext2Lab}$ and $\texttt{Lab2Ext}$ are bijections and each other's inverse.

Proof. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*, and $\mathcal{E}$ and $\mathcal{L}$ be respectively a β-complete extension and a β-complete labeling of $\mathcal{B}$. It suffices to show that $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L})) = \mathcal{L}$ and $\texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E})) = \mathcal{E}$:

a) we will prove $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L})) = \mathcal{L}$:
   - if $\mathcal{L}(A) = \texttt{in} \Rightarrow A \in \texttt{Lab2Ext}(\mathcal{L}) \Rightarrow \texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))(A) = \texttt{in}$;
   - if $\mathcal{L}(A) = \texttt{out} \Rightarrow$ (Proposition 20) $A \in \texttt{in}(\mathcal{L})^+ \Rightarrow A \in \texttt{Lab2Ext}(\mathcal{L})^+ \Rightarrow \texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))(A) = \texttt{out}$;
   - if $\mathcal{L}(A) = \texttt{undec} \Rightarrow A \notin \texttt{in}(\mathcal{L})$ and $A \notin \texttt{out}(\mathcal{L}) \Rightarrow$ (Proposition 20) $A \notin \texttt{in}(\mathcal{L})$ and $A \notin \texttt{in}(\mathcal{L})^+ \Rightarrow A \notin (\texttt{Lab2Ext}(\mathcal{L}) \cup \texttt{Lab2Ext}(\mathcal{L})^+) \Rightarrow$ we have that $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))(A) = \texttt{undec}$;

b) we will prove $\texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E})) = \mathcal{E}$:
   $A \in \texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E}))$ iff $\texttt{Ext2Lab}(\mathcal{E})(A) = \texttt{in}$ iff $A \in \mathcal{E}$.

□

Theorem 24. Let $\mathcal{B}$ be a *BAF*. Then the following holds:

a) $\mathcal{L}$ is a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable labeling of $\mathcal{B}$ iff $\texttt{Lab2Ext}(\mathcal{L})$ is respectively a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable extension of $\mathcal{B}$;

b) $\mathcal{E}$ is a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable extension of $\mathcal{B}$ iff $\texttt{Ext2Lab}(\mathcal{E})$ is respectively a β-complete, β-grounded, β-preferred, β-stable, β-semi-stable labeling of $\mathcal{B}$.

Proof. We prove both directions for the β-complete case:

a) we start by proving that if $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$, then $\texttt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$.

   It suffices to show $\texttt{in}(\mathcal{L}) = \texttt{Lab2Ext}_{\mathcal{B}}(\mathcal{L})$ is a fixpoint of $F_{\mathcal{B}}$ and $\texttt{in}(\mathcal{L})$ is conflict-free:

   $A \in \texttt{in}(\mathcal{L})$ iff (Proposition 20) $\texttt{in}(\mathcal{L})$ defends A iff $A \in F_{\mathcal{B}}(\texttt{in}(\mathcal{L}))$.

   By absurd, suppose $\texttt{in}(\mathcal{L})$ is not conflict-free. This implies there exists an argument $A \in \texttt{in}(\mathcal{L})$ such that $\texttt{in}(\mathcal{L})$ defeats A, i.e., for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in Att(A')$ such that $B \in \mathtt{in}(\mathcal{L})$. It is an absurd as $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathcal{L}(A) = \mathtt{in}$;

b) now we prove that if $\mathcal{E}$ is a β-complete extension of $\mathcal{B}$, then $\mathtt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$.

Let $\mathcal{L} = \mathtt{Ext2Lab}(\mathcal{E})$. For any argument $A \in \mathcal{A}$, the following holds:

- $\mathcal{L}(A) = \mathtt{in}$ iff $A \in \mathcal{E}$ iff $A \in F_{\mathcal{B}}(\mathcal{E})$ iff A is defended by $\mathcal{E}$ in $\mathcal{B}$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in Att(A')$, it holds $B \in \mathcal{E}^+$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for all $B \in Att(A')$, it holds $\mathcal{L}(B) = \mathtt{out}$;
- If $\mathcal{L}(A) = \mathtt{out}$ iff $A \in \mathcal{E}^+$ iff A is defeated by $\mathcal{E}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in Att(A')$ such that $B \in \mathcal{E}$ iff for every $A' \in \mathfrak{Sup}(A)$, there exists an argument $B \in Att(A')$ such that $\mathcal{L}(B) = \mathtt{in}$;

c) $\mathtt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B} \Longrightarrow$ by item b), $\mathtt{Ext2Lab}(\mathtt{Lab2Ext}(\mathcal{L}))$ is a β-complete labeling of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$.

d) $\mathtt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B} \Longrightarrow$ by item a), $\mathtt{Lab2Ext}(\mathtt{Ext2Lab}(\mathcal{E}))$ is a β-complete extension of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{E}$ is a β-complete extension of $\mathcal{B}$.

Now we prove for the β-grounded case:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B} \Longrightarrow \mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-minimal among the β-complete labelings of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\mathtt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$ and $\mathtt{Lab2Ext}(\mathcal{L})$ is $\subseteq$-minimal among the β-complete extensions of $\mathcal{B} \Longrightarrow \mathtt{Lab2Ext}(\mathcal{L})$ is a β-grounded extension of $\mathcal{B}$;

b) $\mathcal{E}$ is a β-grounded extension of $\mathcal{B} \Longrightarrow \mathcal{E}$ is a β-complete extension of $\mathcal{B}$ and $\mathcal{E}$ is $\subseteq$-minimal among the β-complete extensions of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\mathtt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathtt{Ext2Lab}(\mathcal{E}))$ is $\subseteq$-minimal among the β-complete labelings of $\mathcal{B} \Longrightarrow \mathtt{Ext2Lab}(\mathcal{E})$ is a β-grounded labeling of $\mathcal{B}$;

c) $\mathtt{Lab2Ext}(\mathcal{L})$ is a β-grounded extension of $\mathcal{B} \Longrightarrow$ by item b), $\mathtt{Ext2Lab}(\mathtt{Lab2Ext}(\mathcal{L}))$ is a β-grounded labeling of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$.

d) $\mathtt{Ext2Lab}(\mathcal{E})$ is a β-grounded labeling of $\mathcal{B} \Longrightarrow$ by item a), $\mathtt{Lab2Ext}(\mathtt{Ext2Lab}(\mathcal{E}))$ is a β-grounded extension of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{E}$ is a β-grounded extension of $\mathcal{B}$.

Now we prove for the β-preferred case:

a) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B} \Longrightarrow \mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\mathtt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$ and $\mathtt{Lab2Ext}(\mathcal{L})$ is $\subseteq$-maximal among the β-complete extensions of $\mathcal{B} \Longrightarrow \texttt{Lab2Ext}(\mathcal{L})$ is a β-preferred extension of $\mathcal{B}$;

b) $\mathcal{E}$ is a β-preferred extension of $\mathcal{B} \Longrightarrow \mathcal{E}$ is a β-complete extension of $\mathcal{B}$ and $\mathcal{E}$ is $\subseteq$-maximal among the β-complete extensions of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\texttt{Ext2Lab}(\mathcal{E}))$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow \texttt{Ext2Lab}(\mathcal{E})$ is a β-preferred labeling of $\mathcal{B}$;

c) $\texttt{Lab2Ext}(\mathcal{L})$ is a β-preferred extension of $\mathcal{B} \Longrightarrow$ by item b), $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))$ is a β-preferred labeling of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$;

d) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-preferred labeling of $\mathcal{B} \Longrightarrow$ by item a), $\texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E}))$ is a β-preferred extension of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{E}$ is a β-preferred extension of $\mathcal{B}$.

Now we prove for the β-stable case:

a) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B} \Longrightarrow \mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\mathcal{L}) \cup \texttt{out}(\mathcal{L}) = \mathcal{A} \Longrightarrow$ (Proposition 20) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\mathcal{L}) \cup \texttt{in}(\mathcal{L})^{+} = \mathcal{A} \Longrightarrow$ (by the β-complete case) $\texttt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$ and $\texttt{Lab2Ext}(\mathcal{L}) \cup \texttt{Lab2Ext}(\mathcal{L})^{+} = \mathcal{A} \Longrightarrow \texttt{Lab2Ext}(\mathcal{L})$ is a β-stable extension of $\mathcal{B}$;

b) $\mathcal{E}$ is a β-stable extension of $\mathcal{B} \Longrightarrow \mathcal{E}$ is a β-complete extension of $\mathcal{B}$ and $\mathcal{E} \cup \mathcal{E}^{+} = \mathcal{A} \Longrightarrow$ (by the β-complete case) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\texttt{Ext2Lab}(\mathcal{E})) \cup \texttt{in}(\texttt{Ext2Lab}(\mathcal{E}))^{+} = \mathcal{A} \Longrightarrow$ (Proposition 20) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\texttt{Ext2Lab}(\mathcal{E})) \cup \texttt{out}(\texttt{Ext2Lab}(\mathcal{E})) = \mathcal{A} \Longrightarrow \texttt{undec}(\texttt{Ext2Lab}(\mathcal{E})) = \emptyset \Longrightarrow \texttt{Ext2Lab}(\mathcal{E})$ is a β-stable labeling of $\mathcal{B}$;

c) $\texttt{Lab2Ext}(\mathcal{L})$ is a β-stable extension of $\mathcal{B} \Longrightarrow$ by item b), $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))$ is a β-stable labeling of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$.

d) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-stable labeling of $\mathcal{B} \Longrightarrow$ by item a), $\texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E}))$ is a β-stable extension of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{E}$ is a β-stable extension of $\mathcal{B}$.

Now we prove for the β-semi-stable case:

a) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B} \Longrightarrow \mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\mathcal{L}) \cup \texttt{out}(\mathcal{L})$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow$ (Proposition 20) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\mathcal{L}) \cup \texttt{in}(\mathcal{L})^{+}$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\texttt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$ and $\texttt{Lab2Ext}(\mathcal{L}) \cup \texttt{Lab2Ext}(\mathcal{L})^{+}$ is $\subseteq$-maximal among the β-complete extensions of $\mathcal{B} \Longrightarrow \texttt{Lab2Ext}(\mathcal{L})$ is a β-semi-stable extension of $\mathcal{B}$;

b) $\mathcal{E}$ is a β-semi-stable extension of $\mathcal{B} \Longrightarrow \mathcal{E}$ is a β-complete extension of $\mathcal{B}$ and $\mathcal{E} \cup \mathcal{E}^{+}$

is $\subseteq$-maximal among the β-complete extensions of $\mathcal{B} \Longrightarrow$ (by the β-complete case) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\texttt{Ext2Lab}(\mathcal{E})) \cup \texttt{in}(\texttt{Ext2Lab}(\mathcal{E}))^+$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow$ (Proposition 20) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-complete labeling of $\mathcal{B}$ and $\texttt{in}(\texttt{Ext2Lab}(\mathcal{E})) \cup \texttt{out}(\texttt{Ext2Lab}(\mathcal{E}))$ is $\subseteq$-maximal among the β-complete labelings of $\mathcal{B} \Longrightarrow \texttt{Ext2Lab}(\mathcal{E})$ is a β-semi-stable labeling of $\mathcal{B}$;

c) $\texttt{Lab2Ext}(\mathcal{L})$ is a β-semi-stable extension of $\mathcal{B} \Longrightarrow$ by item b), $\texttt{Ext2Lab}(\texttt{Lab2Ext}(\mathcal{L}))$ is a β-semi-stable labeling of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$;

d) $\texttt{Ext2Lab}(\mathcal{E})$ is a β-semi-stable labeling of $\mathcal{B} \Longrightarrow$ by item a), $\texttt{Lab2Ext}(\texttt{Ext2Lab}(\mathcal{E}))$ is a β-semi-stable extension of $\mathcal{B} \Longrightarrow$ (Theorem 23) $\mathcal{E}$ is a β-semi-stable extension of $\mathcal{B}$.

□

Theorem 25. For any *BAF* $\mathcal{B} = (\mathcal{A}, Att, Sup)$, there always exists a β-complete labeling of $\mathcal{B}$.

Proof. it follows directly from Theorems 16 and 24. □

Theorem 26. Let $\mathcal{B}$ be a *BAF*. The β-complete labeling $\mathcal{L}$ of $\mathcal{B}$ where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ is unique.

Proof. It follows from Theorem 24 that $\texttt{Lab2Ext}(\mathcal{L})$ is a β-complete extension of $\mathcal{B}$ where $\texttt{Lab2Ext}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete extensions of $\mathcal{B}$. From Theorem 17, we know $\texttt{Lab2Ext}(\mathcal{L})$ is the unique $\subseteq$-minimal β-complete extension of $\mathcal{B}$. Hence, taking into account Lemma 29, we obtain $\mathcal{L}$ is the unique $\subseteq$-minimal β-complete labeling of $\mathcal{B}$. □

Theorem 30. Let $\mathcal{B}$ be a *BAF*. The following statements are equivalent:

a) $\mathcal{L}$ is a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$;

b) $\mathcal{L}$ is a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$;

c) $\mathcal{L}$ is a β-complete labeling where $\texttt{undec}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

Proof. We prove each direction:

a) direction from a) to b). Let $\mathcal{L}$ be a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$. By absurd, suppose $\texttt{out}(\mathcal{L})$ is not $\subseteq$-minimal. Then there exists a β-complete labeling $\mathcal{L}'$ such that $\texttt{out}(\mathcal{L}') \subset \texttt{out}(\mathcal{L})$. From Lemma 27, it follows $\texttt{in}(\mathcal{L}') \subset \texttt{in}(\mathcal{L})$. It is an absurd as $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal;

b) direction from b) to a). Let $\mathcal{L}$ be a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$. By absurd, suppose $\texttt{in}(\mathcal{L})$ is not $\subseteq$-minimal. Then there exists a β-complete labeling $\mathcal{L}'$ such that $\texttt{in}(\mathcal{L}') \subset \texttt{in}(\mathcal{L})$. From Lemma 27, it follows $\texttt{out}(\mathcal{L}') \subset \texttt{out}(\mathcal{L})$. It is an absurd as $\texttt{out}(\mathcal{L})$ is $\subseteq$-minimal;

c) direction from a) to c). Let $\mathcal{L}$ be a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$. From Theorem 26, it is unique, i.e., $\texttt{in}(\mathcal{L}) \subseteq \texttt{in}(\mathcal{L}')$ for every β-complete labeling $\mathcal{L}'$ of $\mathcal{B}$. By Lemma 28, $\texttt{undec}(\mathcal{L}') \subseteq \texttt{undec}(\mathcal{L})$ for every β-complete labeling $\mathcal{L}'$ of $\mathcal{B}$. Hence, $\mathcal{L}$ is a β-complete labeling where $\texttt{undec}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$;

d) direction from c) to a). Let $\mathcal{L}$ be a β-complete labeling where $\texttt{undec}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$. In order to show that $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal, we have to prove that for each β-complete argument labeling $\mathcal{L}'$, if $\texttt{in}(\mathcal{L}') \subseteq \texttt{in}(\mathcal{L})$, then $\texttt{in}(\mathcal{L}') = \texttt{in}(\mathcal{L})$. Let $\mathcal{L}'$ be a β-complete labeling such that $\texttt{in}(\mathcal{L}') \subseteq \texttt{in}(\mathcal{L})$. From Lemma 28, we obtain that $\texttt{undec}(\mathcal{L}) \subseteq \texttt{undec}(\mathcal{L}')$. As $\texttt{undec}(\mathcal{L})$ is $\subseteq$-maximal, we know $\texttt{undec}(\mathcal{L}) = \texttt{undec}(\mathcal{L}')$. But then, $\texttt{in}(\mathcal{L}') \not\subset \texttt{in}(\mathcal{L})$. Otherwise, $\texttt{undec}(\mathcal{L}) \subset \texttt{undec}(\mathcal{L}')$ would follow from Lemma 28, which contradicts $\texttt{undec}(\mathcal{L}) = \texttt{undec}(\mathcal{L}')$. This implies $\texttt{in}(\mathcal{L}) = \texttt{in}(\mathcal{L}')$ and as a consequence, $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal.

□

Theorem 31. Let $\mathcal{B}$ be a *BAF*. It holds $\mathcal{L}$ is a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

Proof. We prove each direction:

($\Rightarrow$) let $\mathcal{L}$ be a β-complete labeling where $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$. By absurd, suppose $\texttt{out}(\mathcal{L})$ is not $\subseteq$-maximal. Then there exists a β-complete labeling $\mathcal{L}'$ such that $\texttt{out}(\mathcal{L}) \subset \texttt{out}(\mathcal{L}')$. From Lemma 27, it follows $\texttt{in}(\mathcal{L}) \subset \texttt{in}(\mathcal{L}')$. It is an absurd as $\texttt{in}(\mathcal{L})$ is $\subseteq$-maximal;

($\Leftarrow$) let $\mathcal{L}$ be a β-complete labeling where $\texttt{out}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$. By absurd, suppose $\texttt{in}(\mathcal{L})$ is not $\subseteq$-maximal. Then there exists a β-complete labeling $\mathcal{L}'$ such that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$. From Lemma 27, it follows $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$. It is an absurd as $\mathtt{out}(\mathcal{L})$ is $\subseteq$-maximal.

□

Theorem 32. Let $\mathcal{B}$ be a *BAF*. If $\mathcal{L}$ is a β-complete labeling where $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$, then $\mathtt{in}(\mathcal{L})$ and $\mathtt{out}(\mathcal{L})$ are $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$.

Proof. Let $\mathcal{L}$ be a β-complete labeling where $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$. By absurd, suppose $\mathtt{in}(\mathcal{L})$ is not $\subseteq$-maximal (resp. $\mathtt{out}(\mathcal{L})$ is not $\subseteq$-maximal). Then there exists a β-complete labeling $\mathcal{L}'$ such that $\mathtt{in}(\mathcal{L}) \subset \mathtt{in}(\mathcal{L}')$ (resp. $\mathtt{out}(\mathcal{L}) \subset \mathtt{out}(\mathcal{L}')$). From Lemma 28, it follows $\mathtt{undec}(\mathcal{L}') \subset \mathtt{undec}(\mathcal{L}')$. It is an absurd as $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal.

□

Theorems and proofs from Chapter 4.

Theorems and proofs from Section 4.2.

Lemma 34. Let P be an *NLP*, $\mathcal{I}$ an interpretation, and $\Psi^{\uparrow\omega}_{P/\mathcal{I}}$ the least 3-valued model of $P/\mathcal{I}$ (see Definition 22). The following holds:

a) $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff there exists an argument A constructed from P such that $\mathtt{Conc}(A) = c$ and $\mathtt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$;

b) $c \in \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff for each argument A constructed from P such that $\mathtt{Conc}(A) = c$, we have $\mathtt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) \neq \emptyset$.

Proof. Before the proof, we recall some relevant definitions. Define $h^{(0)} = (\emptyset, \mathit{HB}_{P/\mathcal{I}}, \emptyset)$ and $h^{(i+1)} = \Psi_{P/\mathcal{I}}(h^{(i)})$, where Ψ is Przymusinski's (1990) operator (Definition 22). Recall that $P/\mathcal{I}$ is a positive program (see Definition 21). By Definition 22, given $h^{(i)}$ and some atom $c \in \mathit{HB}_P$ (also in $\mathit{HB}_{P/\mathcal{I}}$), we compute $h^{(i+1)}(c)$ as follows:

a) if there is $r \in P/\mathcal{I}$ such that $\mathit{head}(r) = c$ and $\mathit{body}(r) \subseteq \mathbf{t}(h^{(i)})$, then $c \in \mathbf{t}(h^{(i+1)})$;

b) if for every $r \in P/\mathcal{I}$ such that $\mathit{head}(r) = c$, it holds $\mathit{body}(r) \cap \mathbf{f}(h^{(i)}) \neq \emptyset$, then $c \in \mathbf{f}(h^{(i+1)})$;

c) otherwise, $c \in \mathbf{u}(h^{(i+1)})$.

Przymusinski (1990) proved that $h^{(\omega)}$ is the least 3-valued model of $P/\mathcal{I}$, where $\mathbf{t}(h^{(\omega)}) = \bigcup_{i<\omega} \mathbf{t}(h^{(i)})$ and $\mathbf{f}(h^{(\omega)}) = \bigcap_{i<\omega} \mathbf{f}(h^{(i)})$. In the proof below, we prove some properties of $h^{(i)}$ for every $i \in \mathbb{N}$.

Now we are ready to prove each item:

a) proving that $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$:

– ($\Longrightarrow$) consider $h^{(i)}$ for each $i \in \mathbb{N}$. It suffices to prove by induction on the value of i that if $c \in \mathbf{t}(h^{(i)})$, then there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$.

Basis: for $i = 0$, the result is trivial as $\mathbf{t}(h^{(0)}) = \emptyset$.

Step: assume that for every $c' \in \mathbf{t}(h^{(n)})$, there exists an argument A' constructed from P such that $\texttt{Conc}(A') = c'$ and $\texttt{Vul}(A') \subseteq \mathbf{f}(\mathcal{I})$. We will prove that if $c \in \mathbf{t}(h^{(n+1)})$, there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$. If $c \in \mathbf{t}(h^{(n+1)})$, there exists a rule $r = c \leftarrow a_1, \ldots, a_m \in P/\mathcal{I}$ such that $\{a_1, \ldots, a_m\} \subseteq \mathbf{t}(h^{(n)})$. By Definition 21, r comes from a rule r' in P. As no atom $a_1, \ldots, a_m$ is undecided w.r.t. $h^{(m)}$, they are not the atom **u** (always undecided). Hence, r comes from a rule $r' = c \leftarrow a_1, \ldots, a_m, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n \in P$ such that $\{b_1, \ldots, b_n\} \subseteq \mathbf{f}(\mathcal{I})$, i.e., r is obtained by removing all negative literals of r'. It follows via the inductive step that for every $j \in \{1, \ldots, m\}$, there exists an argument A_j constructed from P such that $\texttt{Conc}(A_j) = a_j$ and $\texttt{Vul}(A_j) \subseteq \mathbf{f}(\mathcal{I})$. But then, we can construct from P an argument A with $\texttt{Conc}(A) = c$ where $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \cdots \cup \texttt{Vul}(A_m) \cup \{b_1, \ldots, b_n\}$. This implies that $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$;

– ($\Longleftarrow$) we will prove by structural induction on the construction of arguments that for each argument A constructed from P such that $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$, it follows $\texttt{Conc}(A) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$.

Basis: let A be an argument $c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ ($n \geq 0$) with $\{b_1, \ldots, b_n\} = \texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$. It follows the fact $c \in P/\mathcal{I}$. Then $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$.

Step: assume $A_1, \ldots, A_m$ ($m \geq 1$) are arbitrary arguments constructed from P such that for each $i \in \{1, \ldots, m\}$, if $\texttt{Vul}(A_i) \subseteq \mathbf{f}(\mathcal{I})$, then $\texttt{Conc}(A_i) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$. We will prove that if A is an argument $c \leftarrow (A_1), \ldots, (A_m), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ ($n \geq 0$) constructed from P such that $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$, then $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$:

Let A be such an argument. By Definition 33, there exists a rule $c \leftarrow a_1, \ldots, a_m$, $\texttt{not}\ b_1, \ldots, \texttt{not}\ b_n \in P$ such that $\texttt{Conc}(A_i) = a_i$ for each $i \in \{1, \ldots, m\}$ and $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \cdots \cup \texttt{Vul}(A_m) \cup \{b_1, \ldots, b_n\}$. As $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$, we obtain $\{b_1, \ldots, b_n\} \subseteq \mathbf{f}(\mathcal{I})$ and $\texttt{Vul}(A_i) \subseteq \mathbf{f}(\mathcal{I})$ for each $i \in \{1, \ldots, m\}$. By inductive hypothesis, it follows $\{a_1, \ldots, a_m\} \subseteq \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$. Then $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$;

b) proving that $c \in \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ iff for every argument A constructed from P such that $\texttt{Conc}(A) = c$, we have $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) \neq \emptyset$:

- ($\Longrightarrow$) the desired result follows if we prove by structural induction on the construction of arguments that for each argument A constructed from P such that $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$, it holds $\texttt{Conc}(A) \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$.

  Basis: let A be an argument $c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ $(n \geq 0)$ such that $\{b_1, \ldots, b_n\} \cap \mathbf{t}(\mathcal{I}) = \emptyset$. It follows the fact $c \in P/\mathcal{I}$ or $c \leftarrow \mathbf{u} \in P/\mathcal{I}$. Then $c \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$.

  Step: assume $A_1, \ldots, A_m$ $(m \geq 1)$ are arbitrary arguments constructed from P such that for each $i \in \{1, \ldots, m\}$, if $\texttt{Vul}(A_i) \cap \mathbf{t}(\mathcal{I}) = \emptyset$, then $\texttt{Conc}(A_i) \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$. We will prove that if A is an argument $c \leftarrow (A_1), \ldots, (A_m), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ $(n \geq 0)$ constructed from P such that $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$, then $c \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$:

  Let A be such an argument. By Definition 33, there exists a rule $c \leftarrow a_1, \ldots, a_m, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n \in P$ such that $\texttt{Conc}(A_i) = a_i$ for each $i \in \{1, \ldots, m\}$ and $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \cdots \cup \texttt{Vul}(A_m) \cup \{b_1, \ldots, b_n\}$. As $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$, we obtain $\{b_1, \ldots, b_n\} \cap \mathbf{t}(\mathcal{I}) = \emptyset$ and $\texttt{Vul}(A_i) \cap \mathbf{t}(\mathcal{I}) = \emptyset$ for each $i \in \{1, \ldots, m\}$. By inductive hypothesis, it follows $\{a_1, \ldots, a_m\} \cap \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}}) = \emptyset$. Then, $c \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$;

- ($\Longleftarrow$) assume that for every argument A constructed from P such that $\texttt{Conc}(A) = c$, we have $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) \neq \emptyset$. The proof is by contradiction: suppose that $c \notin \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$.

  Consider $h^{(i)}$ for each $i \in \mathbb{N}$. It suffices to prove by induction on the value of i that if $c \notin \mathbf{f}(h^{(i)})$, then there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$:

  Basis: for $i = 0$, the result is trivial as $\mathbf{f}(h^{(0)}) = HB_P$.

  Step: assume that for every $c' \notin \mathbf{f}(h^{(n)})$, there exists an argument A' constructed from P such that $\texttt{Conc}(A') = c'$ and $\texttt{Vul}(A') \cap \mathbf{t}(\mathcal{I}) = \emptyset$. We will prove that if $c \notin \mathbf{f}(h^{(n+1)})$, there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$:

  If $c \notin \mathbf{f}(h^{(n+1)})$, then there exists a rule $r \in P/\mathcal{I}$ such that $r = c \leftarrow a_1, \ldots, a_m$ with $\{a_1, \ldots, a_m\} \cap \mathbf{f}(h^{(n)}) = \emptyset$. By Definition 21, this means r comes from a rule $r' \in P$ such that $r' = c \leftarrow a_1, \ldots, a_m, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ with $\{b_1, \ldots, b_n\} \cap \mathbf{t}(\mathcal{I}) = \emptyset$. It follows via the inductive step that for every $j \in \{1, \ldots, m\}$, there exists an argument A_j constructed from P such that $\texttt{Conc}(A_j) = a_j$ and $\texttt{Vul}(A_j) \cap \mathbf{t}(\mathcal{I}) = \emptyset$. But then, we can construct from P an argument A with $\texttt{Conc}(A) = c$ where $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \cdots \cup \texttt{Vul}(A_m) \cup \{b_1, \ldots, b_n\}$, which implies $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$.

□

Corollary 35. Let P be an *NLP* and $c \in HB_P$. The two statements below hold:

a) $c \in \mathbf{t}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ for $\mathcal{I} = (\emptyset, HB_P, \emptyset)$ iff there exists an argument A constructed from P such that $\texttt{Conc}(A) = c$;

b) $c \in \mathbf{f}(\Psi^{\uparrow\omega}_{P/\mathcal{I}})$ for every interpretation $\mathcal{I}$ iff there is no argument A constructed from P such that $\texttt{Conc}(A) = c$.

Proof. It follows directly from Lemma 34, as for any argument A constructed from P we have $\texttt{Vul}(A) \subseteq \mathbf{f}(\mathcal{I})$ and $\texttt{Vul}(A) \cap \mathbf{t}(\mathcal{I}) = \emptyset$. □

Theorem 36. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$ and $\texttt{Conc}(A) = \texttt{Conc}(B)$, then $\mathcal{L}(A) = \mathcal{L}(B)$.

Proof. Note $\mathfrak{Sup}(A) = \{A' \in \mathcal{A}_P \mid \texttt{Conc}(A') = \texttt{Conc}(A)\} = \{A' \in \mathcal{A}_P \mid \texttt{Conc}(A') = \texttt{Conc}(B)\} = \mathfrak{Sup}(B)$. The following holds:

a) $\mathcal{L}(A) = \texttt{in}$ iff (Proposition 19) $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A') = \texttt{in}$ for some $A' \in \mathfrak{Sup}(B)$ iff (Proposition 19) $\mathcal{L}(B) = \texttt{in}$;

b) $\mathcal{L}(A) = \texttt{out}$ iff (Proposition 19) $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A)$ iff $\mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(B)$ iff (Proposition 19) $\mathcal{L}(B) = \texttt{out}$.

□

Theorem 37. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. For any β-complete labeling $\mathcal{L}$ of $\mathcal{B}_P$, it holds $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) = \mathcal{L}$.

Proof. Let $A \in \mathcal{A}_P$ and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}) = (T, F, U)$; there are three possibilities:

a) $\mathcal{L}(A) = \texttt{in} \Rightarrow \texttt{Conc}(A) \in T \Rightarrow \mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))(A) = \texttt{in}$;

b) $\mathcal{L}(A) = \texttt{out} \Rightarrow \mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = \texttt{Conc}(A)$ (Theorem 36) $\Rightarrow \texttt{Conc}(A) \in F \Rightarrow \mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))(A) = \texttt{out}$;

c) $\mathcal{L}(A) = \texttt{undec} \Rightarrow \mathcal{L}(A') = \texttt{undec}$ for every $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = \texttt{Conc}(A)$ (Theorem 36) $\Rightarrow \texttt{Conc}(A) \in U \Rightarrow \mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))(A) = \texttt{undec}$.

□

Theorem 38. Let P be an *NLP*, $\mathcal{B}_P = (\mathcal{A}_P, Att_P)$ be its corresponding *BAF*, and $\mathcal{M}$ be a partial stable model of P. It holds that $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})) = \mathcal{M}$.

Proof. Let $\mathcal{M}$ be a partial stable model of P, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M}))$, and $c \in HB_P$. It suffices to prove the following results:

a) $c \in \mathbf{t}(\mathcal{M})$ iff $c \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$:
   - assume $c \in \mathbf{t}(\mathcal{M})$. As $\Psi^{\uparrow\omega}_{P/\mathcal{M}} = \mathcal{M}$, by Lemma 34, there exists an argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ such that $\texttt{Vul}(A) \subseteq F$. As $\texttt{Conc}(A) = c \in \mathbf{t}(\mathcal{M})$, it follows $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})(A) = \texttt{in}$. Consequently, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M}))(c) = \texttt{in}$, i.e., $c \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$;
   - assume $c \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$, i.e., $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M}))(c) = \texttt{in}$. There is an argument $A \in \mathcal{A}_p$ with $\texttt{Conc}(A) = c$ and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})(A) = \texttt{in}$. It follows $\texttt{Conc}(A) = c \in \mathbf{t}(\mathcal{M})$;

b) $c \in \mathbf{f}(\mathcal{M})$ iff $c \in \mathbf{f}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$:
   - assume $c \notin \mathbf{f}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$. Then there exists an argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ and $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})(A) \neq \texttt{out}$. It follows $\texttt{Conc}(A) = c \notin \mathbf{f}(\mathcal{M})$;
   - assume $c \notin \mathbf{f}(\mathcal{M})$. As $\Psi^{\uparrow\omega}_{P/\mathcal{M}} = \mathcal{M}$, by Lemma 34, there is an argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ such that $\texttt{Vul}(A) \cap T = \emptyset$. As $\texttt{Conc}(A) = c \notin \mathbf{f}(\mathcal{M})$, it holds $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})(A) \neq \texttt{out}$ and $\texttt{Conc}(A) = c \notin \mathbf{f}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})))$.

□

Proposition 39. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, then

$$\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})(c) = \begin{cases} \mathbf{t} & \exists A \in \mathcal{A}_P \textit{ such that } \texttt{Conc}(A) = c \textit{ and } \mathcal{L}(A) = \texttt{in} \\ \mathbf{u} & \exists A \in \mathcal{A}_P \textit{ such that } \texttt{Conc}(A) = c \textit{ and } \mathcal{L}(A) = \texttt{undec} \\ \mathbf{f} & \textit{otherwise.} \end{cases}$$

Proof. Let $c \in HB_P$. The following holds:

a) if there is no $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$, it follows $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})(c) = \mathbf{f}$;

b) if there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$, it follows $\mathcal{L}(A) = \mathcal{L}(A')$ whenever

$\texttt{Conc}(A) = \texttt{Conc}(A')$ (Theorem 36). Thus

$$\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})(\texttt{Conc}(A)) = \begin{cases} \mathbf{t} & \textit{if } \mathcal{L}(A) = \texttt{in} \\ \mathbf{f} & \textit{if } \mathcal{L}(A) = \texttt{out} \\ \mathbf{u} & \textit{if } \mathcal{L}(A) = \texttt{undec}. \end{cases}$$

□

Theorem 40. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P;

b) $\mathcal{M}$ is a partial stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P$.

Proof. We prove each direction:

a) if $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, then $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P:
Let $\mathcal{M} = \mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}) = (T, F, U)$. We will show $\mathcal{M}$ is a partial stable model of P, i.e., $\Psi^{\uparrow\omega}_{P/\mathcal{M}} = (T', F', U') = (T, F, U)$:

- $c \in T$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and $\mathcal{L}(A) = \texttt{in}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) = \texttt{out}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = c$ and for every $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) = \texttt{out}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = c$ and $\texttt{Vul}(A') \subseteq F$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \subseteq F$ iff (Lemma 34) $c \in T'$;
- $c \notin F$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and $\mathcal{L}(A) \neq \texttt{out}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) \neq \texttt{in}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = c$ and for every $B \in \mathit{Att}(A')$, it holds $\mathcal{L}(B) \neq \texttt{in}$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and there exists $A' \in \mathcal{A}_P$ such that $\texttt{Conc}(A') = c$ and $\texttt{Vul}(A') \cap T = \emptyset$ iff there exists $A \in \mathcal{A}_P$ such that $\texttt{Conc}(A) = c$ and $\texttt{Vul}(A) \cap T = \emptyset$ iff (Lemma 34) $c \notin F'$;

b) if $\mathcal{M}$ is a partial stable model of P, then $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P$:
Let $\mathcal{M} = (T, F, U)$ be a partial stable model of P. Then $\Psi^{\uparrow\omega}_{P/\mathcal{M}} = (T, F, U)$. We will prove $\mathcal{L} = \mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P$. Let A be an argument in $\mathcal{A}_P$. The following holds:

– $\mathcal{L}(A) = \mathtt{in}$ iff $\mathtt{Conc}(A) \in T$ iff (Lemma 34) there is $A' \in \mathcal{A}_P$ with $\mathtt{Conc}(A') = \mathtt{Conc}(A)$ and $\mathtt{Vul}(A') \subseteq F$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in$ *Att*(A'), it holds $\mathcal{L}(B) = \mathtt{out}$;
– $\mathcal{L}(A) \neq \mathtt{out}$ iff $\mathtt{Conc}(A) \notin F$ iff (Lemma 34) there is $A' \in \mathcal{A}_P$ with $\mathtt{Conc}(A') = \mathtt{Conc}(A)$ and $\mathtt{Vul}(A') \cap T = \emptyset$ iff there exists $A' \in \mathfrak{Sup}(A)$ such that for every $B \in$ *Att*(A'), it holds $\mathcal{L}(B) \neq \mathtt{in}$;

c) if $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P, then $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$:
It holds that $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of $P \Rightarrow$ according to item b) above, $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))$ is a β-complete labeling of $\mathcal{B}_P \Rightarrow$ (via Theorem 37) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$;

d) if $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P$, then $\mathcal{M}$ is a partial stable model of P:
It holds that $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}_P \Rightarrow$ according to item a) above, $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M}))$ is a partial stable model of $P \Rightarrow$ (via Theorem 38) $\mathcal{M}$ is a partial stable model of P.

□

Lemma 77. Let P be an *NLP*, $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its associated *BAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}_P$ respecting $\mathfrak{Sup}$, and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}_1) = (T_1, F_1, U_1)$ and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}_2) = (T_2, F_2, U_2)$. The following holds:

a) $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ iff $T_1 \subseteq T_2$;
b) $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$ iff $T_1 = T_2$;
c) $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$ iff $T_1 \subset T_2$.

Proof. We prove each item:

a) We prove each direction:
– ($\Longrightarrow$) suppose $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$. If $c \in T_1$, by Definition 36, there exists $A \in \mathcal{A}_P$ such that $\mathtt{Conc}(A) = c$ and $\mathcal{L}_1(A) = \mathtt{in}$. From our initial assumption, it follows $\mathcal{L}_2(A) = \mathtt{in}$. So, by Definition 36, $c \in T_2$;
– ($\Longleftarrow$) suppose $T_1 \subseteq T_2$. If $\mathcal{L}_1(A) = \mathtt{in}$, by Definition 36, $\mathtt{Conc}(A) \in T_1$. From our initial assumption, it follows $\mathtt{Conc}(A) \in T_2$. So, by Definition 36, there is some argument $A' \in \mathcal{A}_P$ with $\mathtt{Conc}(A') = \mathtt{Conc}(A)$ and $\mathcal{L}_2(A') = \mathtt{in}$. As $\mathcal{L}_2$ respects $\mathfrak{Sup}$, it follows $\mathcal{L}_2(A) = \mathtt{in}$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Lemma 78. Let P be an *NLP*, $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its associated *BAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}_P$ respecting $\mathfrak{Sup}$, and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}_1) = (T_1, F_1, U_1)$ and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}_2) = (T_2, F_2, U_2)$. The following holds:

a) $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$ iff $U_1 \subseteq U_2$;

b) $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2)$ iff $U_1 = U_2$;

c) $\texttt{undec}(\mathcal{L}_1) \subset \texttt{undec}(\mathcal{L}_2)$ iff $U_1 \subset U_2$.

Proof. We prove each item:

a) We prove each direction:
   - ($\Longrightarrow$) Suppose $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$. If $c \in U_1$, by Definition 36, there exists an argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ and $\mathcal{L}_1(A) = \texttt{undec}$, and for every argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ it holds $\mathcal{L}_1(A) \neq \texttt{in}$. By our initial assumption, there exists an argument $A \in \mathcal{A}_P$ with $\texttt{Conc}(A) = c$ and $\mathcal{L}_2(A) = \texttt{undec}$. As $\mathcal{L}_2$ respects $\mathfrak{Sup}$, for every argument $A' \in \mathcal{A}_P$ with $\texttt{Conc}(A') = c$ it holds $\mathcal{L}_2(A') = \texttt{undec}$. So, by Definition 36, $c \in U_2$;
   - ($\Longleftarrow$) Suppose $U_1 \subseteq U_2$. Assume $\mathcal{L}_1(A) = \texttt{undec}$. As $\mathcal{L}_1$ respects $\mathfrak{Sup}$, for every argument $A' \in \mathcal{A}_P$ with $\texttt{Conc}(A') = \texttt{Conc}(A)$ it holds $\mathcal{L}_1(A') = \texttt{undec}$. By Definition 36, it follows $\texttt{Conc}(A) \in U_1$. From our initial assumption, it follows $\texttt{Conc}(A) \in U_2$. As $\mathcal{L}_2$ respects $\mathfrak{Sup}$, for every argument $A' \in \mathcal{A}_P$ with $\texttt{Conc}(A') = \texttt{Conc}(A)$ it holds $\mathcal{L}_2(A') = \texttt{undec}$. In particular, $\mathcal{L}_2(A) = \texttt{undec}$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Theorem 41. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a well-founded model of P;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a regular model of P;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a stable model of P;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is an L-stable model of P.

Proof. Let $\mathcal{L}$ be a labeling of $\mathcal{B}_P$ and $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$. The proof is straightforward:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}_P$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}_P$ iff (Theorem 40, Proposition 43, and Lemma 77) $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P, and there is no partial stable model $\mathcal{M}'$ of P such that $\mathbf{t}(\mathcal{M}') \subset \mathbf{t}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a well-founded model of P;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}_P$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}_P$ iff (Theorem 40, Proposition 43, and Lemma 77) $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P, and there is no partial stable model $\mathcal{M}'$ of P such that $\mathbf{t}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) \subset \mathbf{t}(\mathcal{M}')$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a regular model of P;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$ and $\mathtt{undec}(\mathcal{L}) = \emptyset$ iff (Theorem 40) $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model such that $\mathbf{u}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})) = \emptyset$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a stable model of P;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}_P$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_P$, and $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}_P$ iff (Theorem 40, Proposition 43, and Lemma 78) $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is a partial stable model of P, and there is no partial stable model $\mathcal{M}'$ of P such that $\mathbf{u}(\mathcal{M}') \subset \mathbf{u}(\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L}))$ iff $\mathcal{L}2\mathcal{I}_{P\to\mathcal{B}_P}(\mathcal{L})$ is an L-stable model of P.

□

Corollary 42. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, \mathit{Att}_P, \mathit{Sup}_P)$ be its corresponding *BAF*. The following holds:

a) $\mathcal{M}$ is a well-founded model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-grounded labeling of $\mathcal{B}_P$;

b) $\mathcal{M}$ is a regular model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-preferred labeling of $\mathcal{B}_P$;

c) $\mathcal{M}$ is a stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-stable labeling of $\mathcal{B}_P$;

d) $\mathcal{M}$ is an L-stable model of P iff $\mathcal{I}2\mathcal{L}_{P\to\mathcal{B}_P}(\mathcal{M})$ is a β-semi-stable labeling of $\mathcal{B}_P$.

Proof. These results come from Theorems 38 and 41. □

Theorems and proofs from Section 4.3.

Proposition 43. If $\mathcal{L}$ is a β-complete labeling of $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$, then $\mathcal{L}$ respects $\mathfrak{Sup}$.

Proof. Let $A, A' \in \mathcal{A}$ such that $\mathfrak{Sup}(A) = \mathfrak{Sup}(A')$. The following holds:

a) $\mathcal{L}(A) = \mathtt{in} \iff \exists B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \subseteq \mathtt{out}(\mathcal{L}) \iff \exists B \in \mathfrak{Sup}(A') : \mathit{Att}(B) \subseteq \mathtt{out}(\mathcal{L}) \iff \mathcal{L}(A') = \mathtt{in}$;

b) $\mathcal{L}(A) = \texttt{out} \iff \forall B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \cap \texttt{in}(\mathcal{L}) \neq \emptyset \iff \forall B \in \mathfrak{Sup}(A') : \mathit{Att}(B) \cap \texttt{in}(\mathcal{L}) \neq \emptyset \iff \mathcal{L}(A') = \texttt{out}$.

□

Theorem 44. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. The following holds:

a) for any labeling $\mathcal{L}$ of $\mathcal{B}$ respecting $\mathfrak{Sup}$, it holds $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$;
b) for any interpretation $\mathcal{I}$ of $P_{\mathcal{B}}$, it holds $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I})) = \mathcal{I}$.

Proof. Both results are immediate:

a) let $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$: $\mathcal{L}(A) = \texttt{in} \Rightarrow \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{t}$ $\Rightarrow \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}))(A) = \texttt{in}$; $\mathcal{L}(A) = \texttt{out} \Rightarrow \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{f} \Rightarrow \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}))(A) = \texttt{out}$; $\mathcal{L}(A) = \texttt{undec} \Rightarrow$ then it follows that $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})(\mathfrak{Sup}(A)) = \mathbf{u} \Rightarrow \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}))(A) = \texttt{undec}$;
b) let $\mathcal{I}$ be an interpretation of $P_{\mathcal{B}}$: $\mathcal{I}(\mathfrak{Sup}(A)) = \mathbf{t} \Rightarrow \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I})(A) = \texttt{in} \Rightarrow \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}))(\mathfrak{Sup}(A)) = \mathbf{t}$; $\mathcal{I}(\mathfrak{Sup}(A)) = \mathbf{f} \Rightarrow \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I})(A) = \texttt{out} \Rightarrow \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}))(\mathfrak{Sup}(A)) = \mathbf{f}$; $\mathcal{I}(\mathfrak{Sup}(A)) = \mathbf{u} \Rightarrow$ then it follows $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I})(A) = \texttt{undec} \Rightarrow \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{I}))(\mathfrak{Sup}(A)) = \mathbf{u}$.

□

Lemma 45. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. Let $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$ and $\mathcal{M}$ be an interpretation of $P_{\mathcal{B}}$. If $\mathcal{L} = \mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ or $\mathcal{M} = \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$, then for any $A \in \mathcal{A}$, the following holds:

a) there exists $B \in \mathfrak{Sup}(A)$ such that $\mathit{Att}(B) \subseteq \texttt{out}(\mathcal{L})$ iff there exists $r \in P_{\mathcal{B}}$ with $\mathit{head}(r) = \mathfrak{Sup}(A)$ such that $\{\mathfrak{Sup}(X) \mid \texttt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \subseteq \mathbf{f}(\mathcal{M})$;
b) for every $B \in \mathfrak{Sup}(A)$ it holds $\mathit{Att}(B) \cap \texttt{in}(\mathcal{L}) \neq \emptyset$ iff for every rule $r \in P_{\mathcal{B}}$ with $\mathit{head}(r) = \mathfrak{Sup}(A)$ it holds $\{\mathfrak{Sup}(X) \mid \texttt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \cap \mathbf{t}(\mathcal{M}) \neq \emptyset$.

Proof. It follows straightforwardly from Definitions 37 and 39, as $\mathbf{t}(\mathcal{M}) = \{\mathfrak{Sup}(X) \mid X \in \texttt{in}(\mathcal{L})\}$, $\mathbf{f}(\mathcal{M}) = \{\mathfrak{Sup}(X) \mid X \in \texttt{out}(\mathcal{L})\}$ and $\mathit{body}^-(r_{A,B}) = \{\texttt{not}\ \mathfrak{Sup}(X) \mid X \in \mathit{Att}(B)\}$.

□

Theorem 46. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF*, $P_{\mathcal{B}}$ be its corresponding *NLP*, $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$ and $\mathcal{M}$ be an interpretation of $P_{\mathcal{B}}$. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is a partial stable model of $P_{\mathcal{B}}$.

b) $\mathcal{M}$ is a partial stable model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-complete labeling of $\mathcal{B}$.

Proof. We prove each item:

a) let $\mathcal{I} = \mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ be an interpretation of $P_{\mathcal{B}}$. Recall that the least model of $P_{\mathcal{B}}/\mathcal{I}$ is denoted as $\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}$.

($\Longrightarrow$) Assume $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$. We will prove that $\mathcal{I}$ is a partial stable model of $P_{\mathcal{B}}$ by showing $\mathbf{t}(\mathcal{I}) = \mathbf{t}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}})$ and $\mathbf{f}(\mathcal{I}) = \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}})$. Recall that $\mathit{body}^+(r) = \emptyset$ for every rule r in $P_{\mathcal{B}}$. Hence, any rule r in $P_{\mathcal{B}}/\mathcal{I}$ satisfies $\mathit{body}^+(r) = \emptyset$ or $\mathit{body}^+(r) = \{\mathbf{u}\}$.

$$\begin{aligned}
&\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{I})\\
&\Longleftrightarrow \mathcal{L}(A) = \mathtt{in}\\
&\Longleftrightarrow \exists B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \subseteq \mathtt{out}(\mathcal{L})\\
&\overset{\text{Lemma 45}}{\Longleftrightarrow} \exists r \in P_{\mathcal{B}} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge\\
&\qquad \{\mathfrak{Sup}(X) \mid \mathtt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \subseteq \mathbf{f}(\mathcal{I})\\
&\Longleftrightarrow \exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) = \emptyset\\
&\Longleftrightarrow \exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) \subseteq \mathbf{t}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}})\\
&\Longleftrightarrow \mathfrak{Sup}(A) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}).
\end{aligned}$$

$$\begin{aligned}
&\mathfrak{Sup}(A) \in \mathbf{f}(\mathcal{I})\\
&\Longleftrightarrow \mathcal{L}(A) = \mathtt{out}\\
&\Longleftrightarrow \forall B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \cap \mathtt{in}(\mathcal{L}) \neq \emptyset\\
&\overset{\text{Lemma 45}}{\Longleftrightarrow} \forall r \in P_{\mathcal{B}} : \mathit{head}(r) = \mathfrak{Sup}(A) \to\\
&\qquad \{\mathfrak{Sup}(X) \mid \mathtt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \cap \mathbf{t}(\mathcal{I}) \neq \emptyset\\
&\Longleftrightarrow \neg\exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A)\\
&\Longleftrightarrow \forall r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \to \mathit{body}^+(r) \cap \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}) \neq \emptyset\\
&\Longleftrightarrow \mathfrak{Sup}(A) \in \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}).
\end{aligned}$$

($\Longleftarrow$) Assume $\mathcal{I}$ is a partial stable model of $P_{\mathcal{B}}$, i.e., $\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}} = \mathcal{I}$. We will prove that $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$. Let $A \in \mathcal{A}$.

$$
\begin{aligned}
&\mathcal{L}(A) = \mathtt{in} \\
&\iff \mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{I}) \\
&\iff \mathfrak{Sup}(A) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}) \\
&\iff \exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \land \mathit{body}^{+}(r) \subseteq \mathbf{t}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}) \\
&\iff \exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \land \mathit{body}^{+}(r) = \emptyset \\
&\iff \exists r \in P_{\mathcal{B}} : \mathit{head}(r) = \mathfrak{Sup}(A) \land \\
&\qquad \{\mathfrak{Sup}(X) \mid \mathtt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^{-}(r)\} \subseteq \mathbf{f}(\mathcal{I}) \\
&\overset{\text{Lemma 45}}{\iff} \exists B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \subseteq \mathtt{out}(\mathcal{L}).
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{L}(A) = \mathtt{out} \\
&\iff \mathfrak{Sup}(A) \in \mathbf{f}(\mathcal{I}) \\
&\iff \mathfrak{Sup}(A) \in \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}) \\
&\iff \forall r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \rightarrow \mathit{body}^{+}(r) \cap \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{I}}) \neq \emptyset \\
&\iff \neg\exists r \in P_{\mathcal{B}}/\mathcal{I} : \mathit{head}(r) = \mathfrak{Sup}(A) \\
&\iff \forall r \in P_{\mathcal{B}} : \mathit{head}(r) = \mathfrak{Sup}(A) \rightarrow \\
&\qquad \{\mathfrak{Sup}(X) \mid \mathtt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^{-}(r)\} \cap \mathbf{t}(\mathcal{I}) \neq \emptyset \\
&\overset{\text{Lemma 45}}{\iff} \forall B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \cap \mathtt{in}(\mathcal{L}) \neq \emptyset;
\end{aligned}
$$

b) let $\mathcal{L}' = \mathcal{I}2\mathcal{L}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{M})$ be a labeling of $\mathcal{B}$. Recall that the least model of $P_{\mathcal{B}}/\mathcal{M}$ is denoted as $\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{M}}$, and that $\mathit{body}^{+}(r) = \emptyset$ for every rule r in $P_{\mathcal{B}}$. Hence, any rule r in $P_{\mathcal{B}}/\mathcal{M}$ satisfies $\mathit{body}^{+}(r) = \emptyset$ or $\mathit{body}^{+}(r) = \{\mathbf{u}\}$.
($\Longrightarrow$) Assume $\mathcal{M}$ is a partial stable model of $P_{\mathcal{B}}$, i.e., $\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{M}} = \mathcal{M}$. We will show that $\mathcal{L}'$ is a β-complete labeling of $\mathcal{B}$. Recall that $\mathcal{L}'$ respects $\mathfrak{Sup}$ by $\mathcal{I}2\mathcal{L}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}$'s definition. Let $A \in \mathcal{A}$.

$$\mathcal{L}'(A) = \texttt{in}$$
$$\iff \mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{M})$$
$$\iff \mathfrak{Sup}(A) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$$
$$\iff \exists r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) \subseteq \mathbf{t}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$$
$$\iff \exists r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) = \emptyset$$
$$\iff \exists r \in P_\mathcal{B} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge$$
$$\{\mathfrak{Sup}(X) \mid \texttt{not } \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \subseteq \mathbf{f}(\mathcal{M})$$
$$\overset{\text{Lemma 45}}{\iff} \exists B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \subseteq \texttt{out}(\mathcal{L}').$$

$$\mathcal{L}'(A) = \texttt{out}$$
$$\iff \mathfrak{Sup}(A) \in \mathbf{f}(\mathcal{M})$$
$$\iff \mathfrak{Sup}(A) \in \mathbf{f}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$$
$$\iff \forall r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \to \mathit{body}^+(r) \cap \mathbf{f}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}}) \neq \emptyset$$
$$\iff \neg\exists r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A)$$
$$\iff \forall r \in P_\mathcal{B} : \mathit{head}(r) = \mathfrak{Sup}(A) \to$$
$$\{\mathfrak{Sup}(X) \mid \texttt{not } \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \cap \mathbf{t}(\mathcal{M}) \neq \emptyset$$
$$\overset{\text{Lemma 45}}{\iff} \forall B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \cap \texttt{in}(\mathcal{L}') \neq \emptyset.$$

($\Longleftarrow$) Assume $\mathcal{L}'$ is a β-complete labeling. We will prove that $\mathcal{M}$ is a partial stable model of $P_\mathcal{B}$ by showing that $\mathbf{t}(\mathcal{M}) = \mathbf{t}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$ and $\mathbf{f}(\mathcal{M}) = \mathbf{f}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$.

$$\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{M})$$
$$\iff \mathcal{L}'(A) = \texttt{in}$$
$$\iff \exists B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \subseteq \texttt{out}(\mathcal{L}')$$
$$\overset{\text{Lemma 45}}{\iff} \exists r \in P_\mathcal{B} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge$$
$$\{\mathfrak{Sup}(X) \mid \texttt{not } \mathfrak{Sup}(X) \in \mathit{body}^-(r)\} \subseteq \mathbf{f}(\mathcal{M})$$
$$\iff \exists r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) = \emptyset$$
$$\iff \exists r \in P_\mathcal{B}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \wedge \mathit{body}^+(r) \subseteq \mathbf{t}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}})$$
$$\iff \mathfrak{Sup}(A) \in \mathbf{t}(\Psi^{\uparrow\omega}_{P_\mathcal{B}/\mathcal{M}}).$$

$$
\begin{aligned}
&\mathfrak{Sup}(A) \in \mathbf{f}(\mathcal{M})\\
&\iff \mathcal{L}'(A) = \mathtt{out}\\
&\iff \forall B \in \mathfrak{Sup}(A) : \mathit{Att}(B) \cap \mathtt{in}(\mathcal{L}') \neq \emptyset\\
&\overset{\text{Lemma 45}}{\iff} \forall r \in P_{\mathcal{B}} : \mathit{head}(r) = \mathfrak{Sup}(A) \rightarrow\\
&\qquad \{\mathfrak{Sup}(X) \mid \mathtt{not}\ \mathfrak{Sup}(X) \in \mathit{body}^{-}(r)\} \cap \mathbf{t}(\mathcal{M}) \neq \emptyset\\
&\iff \neg\exists r \in P_{\mathcal{B}}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A)\\
&\iff \forall r \in P_{\mathcal{B}}/\mathcal{M} : \mathit{head}(r) = \mathfrak{Sup}(A) \rightarrow \mathit{body}^{+}(r) \cap \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{M}}) \neq \emptyset\\
&\iff \mathfrak{Sup}(A) \in \mathbf{f}(\Psi^{\uparrow\omega}_{P_{\mathcal{B}}/\mathcal{M}}).
\end{aligned}
$$

□

Lemma 79. Let $\mathcal{B}$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}$ respecting $\mathfrak{Sup}$. The following holds:

a) $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ iff $\mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) \subseteq \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$;

b) $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$ iff $\mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) = \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$;

c) $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$ iff $\mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) \subset \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) ($\Rightarrow$) suppose $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$. If $\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1))$, then by Definition 39, $A \in \mathtt{in}(\mathcal{L}_1)$. From our initial assumption, it follows $A \in \mathtt{in}(\mathcal{L}_2)$. So, by Definition 39, $\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$.

   ($\Leftarrow$) Suppose $\mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) \subseteq \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$. If $A \in \mathtt{in}(\mathcal{L}_1)$, then by Definition 39, $\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1))$. From our initial assumption, we obtain $\mathfrak{Sup}(A) \in \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$. So, by Definition 39, $A \in \mathtt{in}(\mathcal{L}_2)$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Lemma 80. Let $\mathcal{B}$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}$ respecting $\mathfrak{Sup}$. The following holds:

a) $\mathtt{undec}(\mathcal{L}_1) \subseteq \mathtt{undec}(\mathcal{L}_2)$ iff $\mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) \subseteq \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$;

b) $\mathtt{undec}(\mathcal{L}_1) = \mathtt{undec}(\mathcal{L}_2)$ iff $\mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_1)) = \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\rightarrow P_{\mathcal{B}}}(\mathcal{L}_2))$;

c) $\mathtt{undec}(\mathcal{L}_1) \subset \mathtt{undec}(\mathcal{L}_2)$ iff $\mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_1)) \subset \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) ($\Rightarrow$) suppose $\mathtt{undec}(\mathcal{L}_1) \subseteq \mathtt{undec}(\mathcal{L}_2)$. If $\mathfrak{Sup}(A) \in \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_1))$, then by Definition 39, $A \in \mathtt{undec}(\mathcal{L}_1)$. From our initial assumption, it follows $A \in \mathtt{undec}(\mathcal{L}_2)$. So, by Definition 39, $\mathfrak{Sup}(A) \in \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_2))$.
($\Leftarrow$) Suppose $\mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_1)) \subseteq \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_2))$. If $A \in \mathtt{undec}(\mathcal{L}_1)$, then by Definition 39, $\mathfrak{Sup}(A) \in \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_1))$. From our initial assumption, it follows $\mathfrak{Sup}(A) \in \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}_2))$. So, by Definition 39, $A \in \mathtt{undec}(\mathcal{L}_2)$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Theorem 47. Let $\mathcal{B}$ be a *BAF* and $P_\mathcal{B}$ be its corresponding *NLP*. For any labeling $\mathcal{L}$ of $\mathcal{B}$ respecting $\mathfrak{Sup}$, the following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a well-founded model of $P_\mathcal{B}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a regular model of $P_\mathcal{B}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a stable model of $P_\mathcal{B}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is an L-stable model of $P_\mathcal{B}$.

Proof. Let $\mathcal{L}$ be a labeling of $\mathcal{B}$ respecting $\mathfrak{Sup}$. The proof is straightforward:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ iff (Theorem 46, Proposition 43 and Lemma 79) $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a partial stable model of $P_\mathcal{B}$ and there is no partial stable model $\mathcal{M}$ of $P_\mathcal{B}$ such that $\mathbf{t}(\mathcal{M}) \subset \mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L}))$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a well-founded model of $P_\mathcal{B}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$ iff (Theorem 46, Proposition 43 and Lemma 79) $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a partial stable model of $P_\mathcal{B}$ and there is no partial stable model $\mathcal{M}$ of $P_\mathcal{B}$ such that $\mathbf{t}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})) \subset \mathbf{t}(\mathcal{M})$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a regular model of $P_\mathcal{B}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ with $\mathtt{undec}(\mathcal{L}) = \emptyset$ iff (Theorem 46) $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a partial stable model of $P_\mathcal{B}$ with $\mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})) = \emptyset$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_\mathcal{B}}(\mathcal{L})$ is a stable model of $P_\mathcal{B}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ iff (Theorem 46, Proposition 43 and Lemma 80) $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is a partial stable model of $P_{\mathcal{B}}$ and there is no partial stable model $\mathcal{M}$ of $P_{\mathcal{B}}$ such that $\mathbf{u}(\mathcal{M}) \subset \mathbf{u}(\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L}))$ iff $\mathcal{L}2\mathcal{I}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{L})$ is an L-stable model of $P_{\mathcal{B}}$.

□

Corollary 48. Let $\mathcal{B}$ be a *BAF* and $P_{\mathcal{B}}$ be its corresponding *NLP*. The following holds:

a) $\mathcal{M}$ is a well-founded model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-grounded labeling of $\mathcal{B}$;
b) $\mathcal{M}$ is a regular model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-preferred labeling of $\mathcal{B}$;
c) $\mathcal{M}$ is a stable model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-stable labeling of $\mathcal{B}$;
d) $\mathcal{M}$ is a semi-stable model of $P_{\mathcal{B}}$ iff $\mathcal{I}2\mathcal{L}_{\mathcal{B}\to P_{\mathcal{B}}}(\mathcal{M})$ is a β-semi-stable labeling of $\mathcal{B}$.

Proof. These results come from Theorems 44 and 47. □

Theorems and proofs from Section 4.4.

Lemma 49. For any *NLP* P, the corresponding *BAF* $\mathcal{B}_P$ is a $\mathfrak{S}$-*BAF*.

Proof. Denote $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$. From Definition 35, Sup_P is irreflexive and symmetric, as $(A, B) \in Sup_P$ iff $A \neq B$ and $\mathtt{Conc}(A) = \mathtt{Conc}(B)$. Now let $(A, B), (B, C) \in Sup_P$ with $A \neq C$. We conclude $\mathtt{Conc}(A) = \mathtt{Conc}(B) = \mathtt{Conc}(C)$. As $A \neq C$ and $\mathtt{Conc}(A) = \mathtt{Conc}(C)$, it holds $(A, C) \in Sup_P$. We conclude that $\mathcal{B}_P$ is a $\mathfrak{S}$-*BAF*. □

Proposition 50. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF*. If $Sup = \emptyset$, then $\mathcal{B}$ is a *RFBAF*.

Proof. Assume $Sup = \emptyset$. For any $A \in \mathcal{A}$, $\mathfrak{Sup}(A) = \{A\}$. Then, for every $A, B \in \mathcal{A}$, $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ iff $A = B$. As every argument has a distinct set of supporters, there are no redundant arguments and $\mathcal{B}$ is a *RFBAF*. □

Proposition 51. Let P be an *NLP* and $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$ be its corresponding *BAF*. It holds $\mathcal{B}_P$ is of support-guided attacks.

Proof. Let $A, B \in \mathcal{A}_P$ with $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$. Then, $\mathtt{Conc}(A) = \mathtt{Conc}(B)$. From this equality, $\mathtt{Conc}(A) \in \mathtt{Vul}(X) \Leftrightarrow \mathtt{Conc}(B) \in \mathtt{Vul}(X)$ for any $X \in \mathcal{A}_P$, i.e., $(A, X) \in Att_P \Leftrightarrow (B, X) \in Att_P$ for any $X \in \mathcal{A}_P$. □

Lemma 81. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a $\mathfrak{S}$-*BAF*. For every $A, B \in \mathcal{A}$, it holds $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ iff $B \in \mathfrak{Sup}(A)$.

Proof. We prove both directions:

a) ($\Longrightarrow$) assume $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$. We conclude $B \in \mathfrak{Sup}(B) = \mathfrak{Sup}(A)$;

b) ($\Longleftarrow$) assume $B \in \mathfrak{Sup}(A)$. By the symmetry of *Sup*, $A \in \mathfrak{Sup}(B)$. Let $X \in \mathfrak{Sup}(A)$. As $A \in \mathfrak{Sup}(B)$, it holds $X \in \mathfrak{Sup}(B)$. Therefore, $\mathfrak{Sup}(A) \subseteq \mathfrak{Sup}(B)$. Similarly, let $Y \in \mathfrak{Sup}(B)$. As $B \in \mathfrak{Sup}(A)$, it holds $Y \in \mathfrak{Sup}(A)$. Therefore, $\mathfrak{Sup}(B) \subseteq \mathfrak{Sup}(A)$. We conclude $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$.

□

Lemma 82. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a $\mathfrak{S}$-*BAF*. For every $A, B \in \mathcal{A}$, it holds $(A, B) \in Sup$ iff $A \neq B$ and $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$.

Proof. We prove both directions:

a) ($\Longrightarrow$) assume $(A, B) \in Sup$. From Lemma 81, $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and, as *Sup* is irreflexive, $A \neq B$;

b) ($\Longleftarrow$) assume $A \neq B$ and $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$. From Lemma 81, $A \in \mathfrak{Sup}(B)$. As $A \neq B$, there exists a sequence of distinct arguments $(A_0 = A, A_1, \ldots, A_n = B)$ such that $(A_i, A_{i+1}) \in Sup$ for $0 \leq i < n$ with $n > 0$. As $\mathcal{B}$ is a $\mathfrak{S}$-*BAF*, $(A, A_i) \in Sup$ for every $0 < i \leq n$, since $(X, Y), (Y, Z) \in Sup$ with $X \neq Z$ implies $(X, Z) \in Sup$ for any $X, Y, Z \in \mathcal{A}$. In particular, $(A, B) \in Sup$.

□

Theorem 52. Let $\mathcal{B}$ be a *BAF*, $P_{\mathcal{B}}$ be its corresponding *NLP*, and $\mathcal{B}_{P_{\mathcal{B}}}$ be the corresponding *BAF* of $P_{\mathcal{B}}$. It holds $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are isomorphic iff $\mathcal{B}$ is a $\mathfrak{S}^+$-*RFBAF*.

Proof. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ and $\mathcal{B}' = \mathcal{B}_{P_{\mathcal{B}}} = (\mathcal{A}', Att', Sup')$. We prove both directions:

a) ($\Longrightarrow$) assume $\mathcal{B}$ is not a $\mathfrak{S}^+$-*RFBAF*. We will prove that $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are not isomorphic. By Lemma 49 and Proposition 51, $\mathcal{B}_{P_{\mathcal{B}}}$ is a $\mathfrak{S}$-*BAF* of support-guided attacks. If $\mathcal{B}$ is a *RFBAF*, then $\mathcal{B}$ is a $\mathfrak{S}^+$-*RFBAF*, which contradicts our initial assumption. Moreover, if $\mathcal{B}$ is not a $\mathfrak{S}$-*BAF*, then the isomorphism with $\mathcal{B}_{P_{\mathcal{B}}}$ does not hold (Lemma 49). Hence, we consider $\mathcal{B}$ being redundant and of support cliques. There exist redundant arguments $R_0, R_1 \in \mathcal{A}$. Then, $R_0 \neq R_1$ and $\mathfrak{Sup}(R_0) = \mathfrak{Sup}(R_1)$ and $Att(R_0) = Att(R_1)$. We obtain $|\{Att(B) \mid B \in \mathfrak{Sup}(R_0)\}| < |\mathfrak{Sup}(R_0)|$. Intuitively, R_0 and R_1

derive an identical rule, because they have identical sets of supporters and attackers. Note that

$$\begin{aligned}
|\mathcal{A}'| &= |P_{\mathcal{B}}| \\
&= |\{r_{A,B} \mid A \in \mathcal{A}, B \in \mathfrak{Sup}(A)\}| \\
&= \sum_{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\}} |\{\mathit{Att}(B) \mid B \in X\}| \\
&= |\{\mathit{Att}(B) \mid B \in \mathfrak{Sup}(R_0)\}| + \sum_{\substack{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\} \\ X \neq \mathfrak{Sup}(R_0)}} |\{\mathit{Att}(B) \mid B \in X\}| \\
&< |\mathfrak{Sup}(R_0)| + \sum_{\substack{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\} \\ X \neq \mathfrak{Sup}(R_0)}} |\{\mathit{Att}(B) \mid B \in X\}| \\
&\leq |\mathfrak{Sup}(R_0)| + \sum_{\substack{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\} \\ X \neq \mathfrak{Sup}(R_0)}} |X| \\
&= \sum_{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\}} |X| \\
&= |\mathcal{A}|.
\end{aligned}$$

The equality $\sum_{X \in \{\mathfrak{Sup}(A) | A \in \mathcal{A}\}} |X| = |\mathcal{A}|$ comes from $\mathcal{B}$ being of support cliques. As $|\mathcal{A}'| < |\mathcal{A}|$, $\mathcal{B}$ and $\mathcal{B}_{P_{\mathcal{B}}}$ are not isomorphic;

b) ($\Longleftarrow$) assume $\mathcal{B}$ is a $\mathfrak{S}^+$-*RFBAF*. From each argument $A \in \mathcal{A}$ and $B \in \mathfrak{Sup}(A)$, we obtain in $P_{\mathcal{B}}$ a rule $r_{A,B}$, which is an argument of $\mathcal{B}_{P_{\mathcal{B}}}$. Define the function $f : \mathcal{A} \to \mathcal{A}'$ such that $f(A) = r_{A,A}$.

First, we show that f is injective. Let $A, B \in \mathcal{A}$. $f(A) = f(B) \Longrightarrow r_{A,A} = r_{B,B} \Longrightarrow \mathfrak{Sup}(A) = \mathfrak{Sup}(B) \wedge \mathit{Att}(A) = \mathit{Att}(B) \Longrightarrow A = B$. The last step comes from $\mathcal{B}$ not having redundant arguments.

Now we prove that f is surjective. Let $Y \in \mathcal{A}'$. This argument is also a rule in $P_{\mathcal{B}}$, say $r_{A,B}$ for some $A, B \in \mathcal{A}$ such that $B \in \mathfrak{Sup}(A)$. As $\mathcal{B}$ is a $\mathfrak{S}$-*BAF*, from Lemma 81 we obtain $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$. Then, $f(B) = r_{B,B} = r_{A,B} = Y$.

Therefore, f is bijective. It remains to show that it is an isomorphism.

$$\begin{aligned}
(f(A), f(B)) \in Att' &\iff (r_{A,A}, r_{B,B}) \in Att' \\
&\overset{\text{Defs. } f \text{ and } 34}{\iff} \texttt{Conc}(r_{A,A}) \in \texttt{Vul}(r_{B,B}) \\
&\overset{\text{Def. } 37}{\iff} \mathfrak{Sup}(A) \in \{\mathfrak{Sup}(X) \mid X \in Att(B)\} \\
&\overset{\text{Trivial}}{\iff} \exists A' \in Att(B), \mathfrak{Sup}(A') = \mathfrak{Sup}(A) \\
&\overset{\text{Def. } 40}{\iff} \exists A' \in \mathfrak{Sup}(A), (A', B) \in Att \\
&\overset{\text{Def. } 42}{\iff} (A, B) \in Att.
\end{aligned}$$

The last step comes from $\mathcal{B}$ being of support-guided attacks. Now we prove that the structure of the support relation is preserved:

$$\begin{aligned}
(f(A), f(B)) \in Sup' &\overset{\text{Def. } f}{\iff} (r_{A,A}, r_{B,B}) \in Sup' \\
&\overset{\text{Def. } 34}{\iff} r_{A,A} \neq r_{B,B} \wedge \texttt{Conc}(r_{A,A}) = \texttt{Conc}(r_{B,B}) \\
&\overset{f\text{, injective}}{\iff} A \neq B \wedge \texttt{Conc}(r_{A,A}) = \texttt{Conc}(r_{B,B}) \\
&\overset{\text{Defs. } 33 \text{ and } 37}{\iff} A \neq B \wedge \mathfrak{Sup}(A) = \mathfrak{Sup}(B) \\
&\overset{\text{Lemma } 82}{\iff} (A, B) \in Sup.
\end{aligned}$$

□

Lemma 53. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF*, and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. If $HB_P \neq \{head(r) \mid r \in P\}$, then P and $P_{\mathcal{B}_P}$ are not isomorphic.

Proof. Let $\mathfrak{Sup}(A) \in HB_{P_{\mathcal{B}_P}}$ be any atom of $P_{\mathcal{B}_P}$, where A is an argument in $\mathcal{B}_P$. By Definition 37, $r_{A,A} \in P_{\mathcal{B}_P}$ and $head(r_{A,A}) = \mathfrak{Sup}(A)$. Hence, every atom of $P_{\mathcal{B}_P}$ is in the head of some rule. As this property does not hold for P, it follows that P and $P_{\mathcal{B}_P}$ are not isomorphic. □

Lemma 54. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF* and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. If there is some $r \in P$ such that $body^+(r) \neq \emptyset$, then P and $P_{\mathcal{B}_P}$ are not isomorphic.

Proof. By Definition 35, for every rule $r \in P_{\mathcal{B}_P}$ it holds $body^+(r) = \emptyset$. As this property does not hold for P, it follows that P and $P_{\mathcal{B}_P}$ are not isomorphic. □

Theorem 55. Let P be an *NLP*, $\mathcal{B}_P$ be its corresponding *BAF* and $P_{\mathcal{B}_P}$ be the corresponding *NLP* of $\mathcal{B}_P$. It holds P and $P_{\mathcal{B}_P}$ are isomorphic iff P is an *RALP*.

Proof. We prove both directions:

a) ($\Longrightarrow$) assume $HB_P \neq \{head(r) \mid r \in P\}$ or $body^+(r) \neq \emptyset$ for some $r \in P$. By Lemmas 53 and 54, P and $P_{\mathcal{B}_P}$ are not isomorphic;

b) ($\Longleftarrow$) assume $HB_P = \{head(r) \mid r \in P\}$ and $body^+(r) = \emptyset$ for every $r \in P$. Denote $\mathcal{B}_P = (\mathcal{A}_P, Att_P, Sup_P)$. Define $f : HB_P \to HB_{P_{\mathcal{B}_P}}$ such that $f(c) = \{C \in \mathcal{A}_P \mid \texttt{Conc}(C) = c\}$.

First, we show that f is well-defined, i.e., $f(c) \in HB_{P_{\mathcal{B}_P}}$ for any $c \in HB_P$. Let $c \in HB_P$, the set $f(c) = \{C \in \mathcal{A}_P \mid \texttt{Conc}(C) = c\}$ consists of all arguments with conclusion c. By Definition 35, $(C, C') \in Sup_P$ iff $C \neq C'$ and $\texttt{Conc}(C) = \texttt{Conc}(C')$. Therefore, $f(c) = \mathfrak{Sup}(C)$ for any $C \in \mathcal{A}_P$ with $\texttt{Conc}(C) = c$, and it follows $f(c) \in HB_{P_{\mathcal{B}_P}} = \{\mathfrak{Sup}(A) \mid A \in \mathcal{A}_P\}$.

Now we prove that f is injective. Assume $f(a) = f(b)$. Then $\{A \in \mathcal{A}_P \mid \texttt{Conc}(A) = a\} = \{B \in \mathcal{A}_P \mid \texttt{Conc}(B) = b\}$, and we conclude $a = b$.

Now we prove that f is surjective. Let $Y \in HB_{P_{\mathcal{B}_P}}$. Denote $Y = \mathfrak{Sup}(A)$ for some $A \in \mathcal{A}_P$. As $A \in \mathcal{A}_P$, $\texttt{Conc}(A) \in HB_P$. Observe that $f(\texttt{Conc}(A)) = \{C \in \mathcal{A}_P \mid \texttt{Conc}(C) = \texttt{Conc}(A)\} = \mathfrak{Sup}(A) = Y$.

We have shown that f is a bijection. As $body^+(r) = \emptyset$ for every $r \in P$, then, by Definition 35, $r \in P$ iff $r \in \mathcal{A}_P$.

$$
\begin{aligned}
& f(c) \leftarrow \texttt{not } f(b_1), \cdots, \texttt{not } f(b_m) \in P_{\mathcal{B}_P} \\
\Longleftrightarrow\ & \exists C \in \mathcal{A}_P, \exists B \in \mathfrak{Sup}(C) : \\
& \quad r_{C,B} = f(c) \leftarrow \texttt{not } f(b_1), \cdots, \texttt{not } f(b_m) \in P_{\mathcal{B}_P} \\
\Longleftrightarrow\ & \exists B \in \mathcal{A}_P : r_{B,B} = f(c) \leftarrow \texttt{not } f(b_1), \cdots, \texttt{not } f(b_m) \in P_{\mathcal{B}_P} \\
\Longleftrightarrow\ & \exists B \in \mathcal{A}_P : \mathfrak{Sup}(B) \leftarrow \texttt{not } \mathfrak{Sup}(B_1), \cdots, \texttt{not } \mathfrak{Sup}(B_m) \in P_{\mathcal{B}_P} \\
& \quad \text{where } \texttt{Conc}(B) = c, Att_P(B) = \{B_1, \cdots, B_m\} \\
& \quad \text{and } \texttt{Conc}(B_i) = b_i \text{ for } 1 \leq i \leq m \\
\Longleftrightarrow\ & \exists B \in \mathcal{A}_P : \texttt{Conc}(B) = c \wedge Att_P(B) = \{B_1, \cdots, B_m\} \\
& \quad \text{with } \texttt{Conc}(B_i) = b_i \text{ for } 1 \leq i \leq m \\
\Longleftrightarrow\ & c \leftarrow \texttt{not } b_1, \cdots, \texttt{not } b_m \in P.
\end{aligned}
$$

□

Theorems and proofs from Section 4.5.

Lemma 83. Let P be an *NLP*, $\mathcal{B}_P$ its corresponding *BAF* and P^* the corresponding *RALP* of P. For every argument A in $\mathcal{B}_P$, there exists a rule $r \in P^*$ such that *head*$(r) = \texttt{Conc}(A)$ and *body*$(r) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}$.

Proof. We will prove by structural induction that for every argument A in $\mathcal{B}_P$, there exists a rule $r \in P^*$ such that *head*$(r) = \texttt{Conc}(A)$ and *body*$(r) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}$. Let A be an argument in $\mathcal{B}_P$ and assume that for every proper subargument $A' \in \texttt{Sub}(A) - \{A\}$, there exists a rule $r' \in P^*$ such that *head*$(r') = \texttt{Conc}(A')$ and *body*$(r') = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A')\}$.

a) (base) assume A has no proper subarguments, i.e., $\texttt{Sub}(A) = \{A\}$. Then, $A \in P^*$ and *head*$(A) = \texttt{Conc}(A)$ and *body*$(A) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}$;

b) (inductive step) assume A has some proper subargument. Then, $A = a \leftarrow (A_1), \ldots, (A_n), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ $(n > 0)$ derives from a rule $r_0 = a \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ in P with $\texttt{Conc}(A_i) = a_i$ for $1 \leq i \leq n$. By the induction hypothesis, for each A_i, $1 \leq i \leq n$, there exists $r_i \in P^*$ with *head*$(r_i) = \texttt{Conc}(A_i)$ and *body*$(r_i) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A_i)\}$. By Definition 33, $r_0 \notin \texttt{Rules}(A_i)$ for $1 \leq i \leq n$. By Definition 46, there exists $r \in P^*$ with *head*$(r) =$ *head*$(r_0) = a$ and

$$\begin{aligned} \textit{body}(r) &= \bigcup_{i=0}^{n} \textit{body}^-(r_i) \\ &= \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\} \cup \bigcup_{i=1}^{n} \textit{body}^-(r_i) \\ &= \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\} \cup \bigcup_{i=1}^{n} \{\texttt{not}\ x \mid x \in \texttt{Vul}(A_i)\} \\ &= \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}. \end{aligned}$$

□

Lemma 84. Let P be an *NLP*, $\mathcal{B}_P$ its corresponding *BAF* and P^* the corresponding *RALP* of P. For every rule $r \in P^*$, there exists an argument A in $\mathcal{B}_P$ such that *head*$(r) = \texttt{Conc}(A)$ and *body*$(r) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}$.

Proof. We will prove by structural induction that for every rule $r \in P^*$, there exists an argument A in $\mathcal{B}_P$ such that *head*$(r) = \texttt{Conc}(A)$ and *body*$(r) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}$. Let r be a rule in P^* deriving from $r_0 \in P$ and assume that for every proper subrule $r' \in \texttt{Rules}(r) - \{r_0\}$, there exists an argument A' in $\mathcal{B}_P$ such that *head*$(r') = \texttt{Conc}(A')$ and *body*$(r') = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A')\}$:

a) (base) assume r has no proper subrules, i.e., $\texttt{Rules}(r) = \{r_0\}$. Then, $r \in P$ and r is also an argument in $\mathcal{B}_P$ with *head*$(r) = \texttt{Conc}(r)$ and *body*$(r) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(r)\}$;

b) (inductive step) assume r has some proper subrule. Denote $r_0 = a \leftarrow a_1, \ldots, a_n$, $\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ ($n > 0$). By Definition 46, for every a_i, $1 \leq i \leq n$, there exists $r_i \in P^*$ with *head*$(r_i) = a_i$ and $r_0 \notin \texttt{Rules}(r_i)$. By the induction hypothesis, for each r_i, $1 \leq i \leq n$, there exists an argument A_i in $\mathcal{B}_P$ with $\texttt{Conc}(A_i) = a_i$ and *body*$(r_i) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(A_i)\}$. By Definition 33, $\mathcal{B}_P$ has an argument $A = a \leftarrow (A_1), \ldots, (A_n), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ deriving from r_0 with $\texttt{Conc}(A) = a = $ *head*(r) and

$$
\begin{aligned}
\textit{body}(r) &= \bigcup_{i=0}^{n} \textit{body}^-(r_i) \\
&= \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\} \cup \bigcup_{i=1}^{n} \textit{body}^-(r_i) \\
&= \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\} \cup \bigcup_{i=1}^{n} \{\texttt{not}\ x \mid x \in \texttt{Vul}(A_i)\} \\
&= \{\texttt{not}\ x \mid x \in \texttt{Vul}(A)\}.
\end{aligned}
$$

□

Theorem 56. Let P be an *NLP* and P^* be its corresponding *RALP*. It holds P^* and $P_{\mathcal{B}_P}$ are isomorphic.

Proof. Recall that arguments in $\mathcal{B}_P$ are constructed from rules in P (Section 4.2), and atoms in the rules of $P_{\mathcal{B}_P}$ are sets of supporters of arguments in $\mathcal{B}_P$ (Section 4.3). Each atom s in $HB_{P_{\mathcal{B}_P}}$ is a subset of arguments in $\mathcal{B}_P$ with the same conclusion. Thus, we can define the function $f : HB_{P_{\mathcal{B}_P}} \rightarrow HB_{P^*}$ such that $f(s) = c$, where $c = \texttt{Conc}(A)$ for every $A \in s$. As c is the conclusion of some argument A in $\mathcal{B}_P$, then, by Lemma 83, there exists a rule $r \in P^*$ with *head*$(r) = \texttt{Conc}(A) = c$ and therefore $c \in HB_{P^*}$. Hence, f is well-defined.

We will prove that f is injective. Assume $f(s) = f(s') = c$ for $s, s' \in HB_{P_{\mathcal{B}_P}}$. Then, $s = \mathfrak{Sup}(A)$ for some argument A in $\mathcal{B}_P$ with $\texttt{Conc}(A) = c$. Similarly, $s' = \mathfrak{Sup}(A')$ for some argument A' in $\mathcal{B}_P$ with $\texttt{Conc}(A') = c$. As $\texttt{Conc}(A) = \texttt{Conc}(A')$, it follows that $s = \mathfrak{Sup}(A) = \mathfrak{Sup}(A') = s'$.

We will prove that f is surjective. Let $a \in HB_{P^*}$. As P^* is an *RALP*, a is the head of some rule $r \in P^*$. By Lemma 84, $\mathcal{B}_P$ has some argument A with $\texttt{Conc}(A) = a$. Observe that $\mathfrak{Sup}(A) \in HB_{P_{\mathcal{B}_P}}$ and $f(\mathfrak{Sup}(A)) = a$.

We finish this proof by showing that $f(s_0) \leftarrow \texttt{not}\ f(s_1), \ldots, \texttt{not}\ f(s_m) \in P^*$ iff $s_0 \leftarrow \texttt{not}\ s_1, \ldots, \texttt{not}\ s_m \in P_{\mathcal{B}_P}$ for every $s_0, s_1, \ldots, s_m \in HB_{P_{\mathcal{B}_P}}$.

a) ($\Longrightarrow$) let $r = f(\mathfrak{Sup}(B_0)) \leftarrow \texttt{not}\ f(\mathfrak{Sup}(B_1)), \ldots, \texttt{not}\ f(\mathfrak{Sup}(B_m))$ be a rule in P^* (it always follows this format by the proof that f is surjective) for arguments $B_0, B_1, \ldots, B_m$ in $\mathcal{B}_P$ where $\texttt{Conc}(B_i) = b_i = f(\mathfrak{Sup}(B_i))$ for $0 \leq i \leq m$. By applying Lemma 84 to r, $\mathcal{B}_P$ has an argument A with $\texttt{Conc}(A) = \mathit{head}(r) = f(\mathfrak{Sup}(B_0)) = b_0$ and $\texttt{Vul}(A) = \{x \mid \texttt{not}\ x \in \mathit{body}(r)\} = \{f(\mathfrak{Sup}(B_1)), \ldots, f(\mathfrak{Sup}(B_m))\} = \{b_1, \ldots, b_m\}$. Note that $\{\mathfrak{Sup}(X) \mid X \in \mathit{Att}(A)\} = \{\mathfrak{Sup}(B_1), \ldots, \mathfrak{Sup}(B_m)\}$. By Definition 37, $r_{A,A} = \mathfrak{Sup}(A) \leftarrow \texttt{not}\ \mathfrak{Sup}(B_1), \ldots, \texttt{not}\ \mathfrak{Sup}(B_m)$ is a rule in $P_{\mathcal{B}_P}$. As $\texttt{Conc}(A) = \texttt{Conc}(B_0)$, then $\mathfrak{Sup}(A) = \mathfrak{Sup}(B_0)$ and $\mathfrak{Sup}(B_0) \leftarrow \texttt{not}\ \mathfrak{Sup}(B_1), \ldots, \texttt{not}\ \mathfrak{Sup}(B_m) \in P_{\mathcal{B}_P}$;

b) ($\Longleftarrow$) let $r = s_0 \leftarrow \texttt{not}\ s_1, \ldots, \texttt{not}\ s_m$ be a rule in $P_{\mathcal{B}_P}$. By Definition 37, $r = r_{B_0,B_0}$ for some argument B_0 in $\mathcal{B}_P$, i.e., we can write $r = \mathfrak{Sup}(B_0) \leftarrow \texttt{not}\ \mathfrak{Sup}(B_1), \ldots, \texttt{not}\ \mathfrak{Sup}(B_m)$ for arguments $B_0, B_1, \ldots, B_m$ in $\mathcal{B}_P$ where $\texttt{Conc}(B_i) = b_i$ for $0 \leq i \leq m$ and $\texttt{Vul}(B_0) = \{b_1, \ldots, b_m\}$. By applying Lemma 83 to B_0, there is $r_0 \in P^*$ such that $\mathit{head}(r_0) = \texttt{Conc}(B_0) = b_0$ and $\mathit{body}(r_0) = \{\texttt{not}\ x \mid x \in \texttt{Vul}(B_0)\} = \{\texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$. Then,

$$\begin{aligned} r_0 &= b_0 \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m \\ &= f(s_0) \leftarrow \texttt{not}\ f(s_1), \ldots, \texttt{not}\ f(s_1) \in P^*. \end{aligned}$$

□

Corollary 57. Let P be an *NLP* with corresponding *RALP* P^*. It holds $\mathcal{M}$ is a partial stable, well-founded, regular, stable, and L-stable model of P iff $\mathcal{M}$ is respectively a partial stable, well-founded, regular, stable, and L-stable model of P^*.

Proof. It follows from the isomorphism between $P_{\mathcal{B}_P}$ and P^* (Theorem 56), as transforming an *NLP* into a *BAF* (Section 4.2) and a *BAF* into an *NLP* (Section 4.3) preserves the partial stable, well-founded, regular, stable, and L-stable semantics. Each atom a in P is represented by the atom $\mathfrak{Sup}(A)$ in $P_{\mathcal{B}_P}$ for some argument A in $\mathcal{B}_P$ with $\texttt{Conc}(A) = a$. The isomorphism $f : HB_{P_{\mathcal{B}_P}} \to HB_{P^*}$ between $P_{\mathcal{B}_P}$ and P^* is such that $f(\mathfrak{Sup}(A)) = a$ (see proof of Theorem 56). Hence, we obtain the coincidence that $\mathcal{M}$ is a partial stable, well-founded, regular, stable, and L-stable model of P iff $\mathcal{M}$ is also respectively a partial stable, well-founded, regular, stable, and L-stable model of P^*. □

Theorem 58. *NLP*s and *RALP*s have the same expressiveness for partial stable, well-founded, regular, stable, and L-stable semantics.

Proof. The following holds:

a) for any *NLP* P, there is an *RALP* P^* such that $\mathcal{M}$ is a partial stable, well-founded, regular, stable, L-stable model of P iff $\mathcal{M}$ is respectively a partial stable, well-founded, regular, stable, L-stable model of P^* (Corollary 57);
b) obviously, any *RALP* is an *NLP*.

Hence, *NLP*s and *RALP*s have the same expressiveness for the partial stable, well-founded, regular, stable, and L-stable semantics. □

Theorem 59. Let P be an *NLP* and P^* be its corresponding *RALP*. If $\mathcal{B}_P$ is an *RFBAF*, then $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$ are isomorphic.

Proof. By Theorem 56, $P_{\mathcal{B}_P}$ and P^* are isomorphic. Then, $\mathcal{B}_{P_{\mathcal{B}_P}}$ and $\mathcal{B}_{P^*}$ are isomorphic. As $\mathcal{B}_P$ is an *RFBAF*, it is a $\mathfrak{S}$-*RFBAF*. By Theorem 52, $\mathcal{B}_{P_{\mathcal{B}_P}}$ and $\mathcal{B}_P$ are isomorphic. Hence, $\mathcal{B}_P$ and $\mathcal{B}_{P^*}$ are isomorphic. □

Theorem 60. Let P and P' be *RALP*s. It holds P and P' are isomorphic iff $\mathcal{B}_P$ and $\mathcal{B}_{P'}$ are isomorphic.

Proof. ($\Longrightarrow$) Trivial. ($\Longleftarrow$) Assume $\mathcal{B}_P$ and $\mathcal{B}_{P'}$ are isomorphic. Then, $P_{\mathcal{B}_P}$ and $P_{\mathcal{B}_{P'}}$ are isomorphic. As P and P' are *RALP*s, it follows, by Theorem 55, that $P_{\mathcal{B}_P}$ is isomorphic to P, and that $P_{\mathcal{B}_{P'}}$ is isomorphic to P'. Hence, P and P' are isomorphic. □

Theorem 61. Let P_1 and P_2 be *NLP*s with corresponding *RALP*s P_1^* and P_2^*, respectively. If $\mathcal{B}_{P_1}$ and $\mathcal{B}_{P_2}$ are *RFBAF*s, it holds P_1^* and P_2^* are isomorphic iff $\mathcal{B}_{P_1}$ and $\mathcal{B}_{P_2}$ are isomorphic.

Proof. Let $\cong$ be the isomorphism equivalence relation. By Theorem 56, $P_1^* \cong P_{\mathcal{B}_{P_1}}$ and $P_2^* \cong P_{\mathcal{B}_{P_2}}$. Then, $\mathcal{B}_{P_1^*} \cong \mathcal{B}_{P_{\mathcal{B}_{P_1}}}$ and $\mathcal{B}_{P_2^*} \cong \mathcal{B}_{P_{\mathcal{B}_{P_2}}}$. As $\mathcal{B}_{P_1}$ and $\mathcal{B}_{P_2}$ are *RFBAF*s, they are $\mathfrak{S}$-*RFBAF*s. By Theorem 52, $\mathcal{B}_{P_{\mathcal{B}_{P_1}}} \cong \mathcal{B}_{P_1}$ and $\mathcal{B}_{P_{\mathcal{B}_{P_2}}} \cong \mathcal{B}_{P_2}$.

a) ($\Longrightarrow$) assume $P_1^* \cong P_2^*$. Then, $\mathcal{B}_{P_1^*} \cong \mathcal{B}_{P_2^*}$ and $\mathcal{B}_{P_{\mathcal{B}_{P_1}}} \cong \mathcal{B}_{P_{\mathcal{B}_{P_2}}}$. Thus, $\mathcal{B}_{P_1} \cong \mathcal{B}_{P_2}$;
b) ($\Longleftarrow$) assume $\mathcal{B}_{P_1} \cong \mathcal{B}_{P_2}$. Then, $P_{\mathcal{B}_{P_1}} \cong P_{\mathcal{B}_{P_2}}$ and $P_1^* \cong P_2^*$.

□

Theorems and proofs from Chapter 5.

Theorems and proofs from Section 5.2.

Theorem 62. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. For any β-complete labeling $\mathcal{L}$ of $\mathcal{B}$, it holds $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$.

Proof. Let $A \in \mathcal{A}$, $\mathcal{L}' = \text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ and $\mathcal{L}'' = \text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}')$. There are three possibilities:

a) $\mathcal{L}(A) = \texttt{in} \overset{Def.\,49}{\Rightarrow} \mathcal{L}'(\mathfrak{Sup}(A)) = \texttt{in} \overset{Def.\,49}{\Rightarrow} \mathcal{L}''(A) = \texttt{in}$;

b) $\mathcal{L}(A) = \texttt{out} \overset{Prop.\,3}{\Rightarrow} \mathcal{L}(A') = \texttt{out}$ for every $A' \in \mathfrak{Sup}(A) \overset{Def.\,49}{\Rightarrow} \mathcal{L}'(\mathfrak{Sup}(A)) = \texttt{out} \Rightarrow \mathcal{L}''(A) = \texttt{out}$;

c) $\mathcal{L}(A) = \texttt{undec} \overset{Prop.\,3}{\Rightarrow} \mathcal{L}(A') \neq \texttt{out}$ for some $A' \in \mathfrak{Sup}(A)$ and $\mathcal{L}(A') \neq \texttt{in}$ for every $A' \in \mathfrak{Sup}(A) \overset{Def.\,49}{\Rightarrow} \mathcal{L}'(\mathfrak{Sup}(A)) \neq \texttt{out}$ and $\mathcal{L}'(\mathfrak{Sup}(A)) \neq \texttt{in} \Rightarrow \mathcal{L}'(\mathfrak{Sup}(A)) = \texttt{undec} \overset{Def.\,49}{\Rightarrow} \mathcal{L}''(A) = \texttt{undec}$.

□

Theorem 63. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. For any complete labeling $\mathcal{L}$ of $\mathfrak{A}_{\mathcal{B}}$, it holds $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})) = \mathcal{L}$.

Proof. Let $\mathcal{S} \in \mathfrak{S}(\mathcal{A})$. We can write $\mathcal{S} = \mathfrak{Sup}(A)$ for some $A \in \mathcal{A}$. Let $\mathcal{L}' = \text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ and $\mathcal{L}'' = \text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}')$. There are three possibilities:

a) $\mathcal{L}(\mathfrak{Sup}(A)) = \texttt{in} \overset{Def.\,49}{\Rightarrow} \mathcal{L}'(A) = \texttt{in} \overset{Def.\,49}{\Rightarrow} \mathcal{L}''(\mathfrak{Sup}(A)) = \texttt{in}$;

b) $\mathcal{L}(\mathfrak{Sup}(A)) = \texttt{out} \Rightarrow \exists \mathbf{X} \in Att_{\mathcal{B}}(\mathfrak{Sup}(A)), \forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in} \Rightarrow$ there exists a $\subseteq$-minimal set $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\forall B \in \mathfrak{Sup}(A)$, $\mathbf{X} \cap \mathfrak{S}(Att(B)) \neq \emptyset$ and $\forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in}$. As $\mathfrak{Sup}(A') \subseteq \mathfrak{Sup}(A)$ for any $A' \in \mathfrak{Sup}(A)$, it holds $\forall B' \in \mathfrak{Sup}(A)$, there exists a $\subseteq$-minimal set $\mathbf{X} \subseteq \mathfrak{S}(\mathcal{A})$ such that $\forall B'' \in \mathfrak{Sup}(B'), \mathbf{X} \cap \mathfrak{S}(Att(B'')) \neq \emptyset$ and $\forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in} \Rightarrow \forall B' \in \mathfrak{Sup}(A), \exists \mathbf{X} \in Att_{\mathcal{B}}(\mathfrak{Sup}(B')), \forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in} \Rightarrow \forall B' \in \mathfrak{Sup}(A), \mathcal{L}(\mathfrak{Sup}(B')) = \texttt{out} \overset{Def.\,49}{\Rightarrow}$ $\forall B' \in \mathfrak{Sup}(A), \mathcal{L}'(B') = \texttt{out} \overset{Def.\,49}{\Rightarrow} \mathcal{L}''(\mathfrak{Sup}(A)) = \texttt{out}$;

c) the case for $\mathcal{L}(\mathfrak{Sup}(A)) = \texttt{undec}$ follows similarly to the previous item.

□

Lemma 85. Let $\mathbf{H} \subseteq 2^{\mathcal{A}}$ be a collection of sets of arguments. Let $\mathcal{L} : \mathcal{A} \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$ be a labeling. For any $v \in \{\texttt{in}, \texttt{out}, \texttt{undec}\}$, the following holds:

a) $\forall \mathcal{H} \in \mathbf{H}, \exists H \in \mathcal{H}, \mathcal{L}(H) = v$ iff $\exists \mathcal{T} \in \text{Tr}[\mathbf{H}], \forall T \in \mathcal{T}, \mathcal{L}(T) = v$;

b) $\exists \mathcal{H} \in \mathbf{H}, \forall H \in \mathcal{H}, \mathcal{L}(H) = v$ iff $\forall \mathcal{T} \in \text{Tr}[\mathbf{H}], \exists T \in \mathcal{T}, \mathcal{L}(T) = v$.

Proof. We prove each item:

a) we prove each direction:

– ($\Rightarrow$) assume $\forall \mathcal{H} \in \mathbf{H}, \exists H \in \mathcal{H}, \mathcal{L}(H) = v$. Define $\mathcal{T} = \{T \in \mathcal{A} \mid \mathcal{L}(T) = v\}$. By our initial assumption, $\mathcal{T}$ is a transversal of $\mathbf{H}$. By the finiteness of $\mathbf{H}$, there is a minimal transversal $\mathcal{T}^* \in \text{Tr}[\mathbf{H}]$ satisfying $\mathcal{T}^* \subseteq \mathcal{T}$ and thus $\forall T \in \mathcal{T}^*, \mathcal{L}(T) = v$;

– ($\Leftarrow$) assume $\exists \mathcal{T} \in \text{Tr}[\mathbf{H}], \forall T \in \mathcal{T}, \mathcal{L}(T) = v$. Let $\mathcal{H} \in \mathbf{H}$. As $\mathcal{T}$ is a transversal of $\mathbf{H}$, there exists $H \in \mathcal{T} \cap \mathcal{H}$. As $H \in \mathcal{T}$, it holds $\mathcal{L}(H) = v$;

b) we prove each direction:

– ($\Rightarrow$) assume $\exists \mathcal{H} \in \mathbf{H}, \forall H \in \mathcal{H}, \mathcal{L}(H) = v$. Let $\mathcal{T} \in \text{Tr}[\mathbf{H}]$. As $\mathcal{T}$ is a transversal of $\mathbf{H}$ and $\mathcal{H} \in \mathbf{H}$, there is $H \in \mathcal{T} \cap \mathcal{H}$. As $H \in \mathcal{H}$, it follows $\mathcal{L}(H) = v$;

– ($\Leftarrow$) assume $\forall \mathcal{T} \in \text{Tr}[\mathbf{H}], \exists T \in \mathcal{T}, \mathcal{L}(T) = v$. By contradiction, assume $\forall \mathcal{H} \in \mathbf{H}, \exists H \in \mathcal{H}, \mathcal{L}(H) \neq v$. Define $\mathcal{T} = \{T \in \mathcal{A} \mid \mathcal{L}(T) \neq v\}$. It follows that $\mathcal{T}$ is a transversal of $\mathbf{H}$. By the finiteness of $\mathbf{H}$, there exists a minimal transversal $\mathcal{T}^*$ of $\mathbf{H}$ such that $\mathcal{T}^* \subseteq \mathcal{T}$. By the definition of $\mathcal{T}$, we have $\forall T \in \mathcal{T}^*, \mathcal{L}(T) \neq v$. As $\mathcal{T}^* \in \text{Tr}[\mathbf{H}]$, by our initial assumption, we have $\exists T \in \mathcal{T}^*, \mathcal{L}(T) = v$, which is a contradiction.

□

Lemma 86. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_\mathcal{B} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_\mathcal{B})$ be its corresponding *SETAF*. Let $\mathcal{L}'$ be a labeling of $\mathfrak{A}_\mathcal{B}$. For any $v \in \{\texttt{in}, \texttt{out}, \texttt{undec}\}$, the following holds:

a) $\exists \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = v$ iff $\exists \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \forall \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = v$;

b) $\forall \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = v$ iff $\forall \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \exists \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = v$.

Proof. We prove each item:

a) we prove each direction:

– ($\Rightarrow$) assume $\exists \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = v$. Then let $\mathcal{H} = \mathit{Att}(A')$ for some $A' \in \mathfrak{Sup}(A)$ and let $\mathbf{H}' = \mathfrak{S}(\mathit{Att}(A'))$. Clearly, $\mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}$. Let $\mathcal{S} \in \mathbf{H}'$. There exists $B' \in \mathcal{H}$ such that $\mathfrak{Sup}(B') = \mathcal{S}$. By our initial assumption, $\mathcal{L}'(\mathfrak{Sup}(B')) = v$. Then, $\mathcal{L}'(\mathcal{S}) = v$;

– ($\Leftarrow$) assume $\exists \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \forall \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = v$. Then let $\mathbf{H}' = \mathfrak{S}(\mathit{Att}(A'))$ for some $A' \in \mathfrak{Sup}(A)$ and let $\mathcal{H} = \mathit{Att}(A')$. Clearly, $\mathcal{H} \in$

$\{Att(A') \mid A' \in \mathfrak{Sup}(A)\}$. Let $B \in \mathcal{H}$. As $\mathfrak{Sup}(B) \in \mathfrak{S}(\mathcal{H}) = \mathbf{H}'$, it follows from our initial assumption that $\mathcal{L}'(\mathfrak{Sup}(B)) = v$;

b) we prove each direction:

- $(\Rightarrow)$ assume $\forall \mathcal{H} \in \{Att(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = v$. Let $\mathbf{H}' \in \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$. We can write $\mathbf{H}' = \mathfrak{S}(Att(A'))$ for some $A' \in \mathfrak{Sup}(A)$. From our initial assumption, $\exists B \in Att(A'), \mathcal{L}'(\mathfrak{Sup}(B)) = v$. Then, there is $\mathcal{S} = \mathfrak{Sup}(B) \in \mathbf{H}'$ such that $\mathcal{L}'(\mathcal{S}) = v$;
- $(\Leftarrow)$ assume $\forall \mathbf{H}' \in \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}, \exists \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = v$. Let $\mathcal{H} \in \{Att(A') \mid A' \in \mathfrak{Sup}(A)\}$. We can write $\mathcal{H} = Att(A')$ for some $A' \in \mathfrak{Sup}(A)$. From our initial assumption, $\exists \mathcal{S} \in \mathfrak{S}(Att(A')), \mathcal{L}'(\mathcal{S}) = v$. There is $B \in Att(A')$ such that $\mathfrak{Sup}(B) = \mathcal{S}$. Hence, $\exists B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = v$.

□

Theorem 64. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$;

b) $\mathcal{L}$ is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}$.

Proof. We prove each item:

a) if $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$, then $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$, denoted as $\mathcal{L}'$, is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$.

   Assume $\mathcal{L}$ is β-complete. We will show $\mathcal{L}'$ is complete. For any $A \in \mathcal{A}$, the following holds:

$$\begin{aligned}
&\mathcal{L}'(\mathfrak{Sup}(A)) = \texttt{in} \\
&\overset{\text{Def. 49}}{\Leftrightarrow} \exists A' \in \mathfrak{Sup}(A), \mathcal{L}(A') = \texttt{in} \\
&\overset{\text{Prop. 3}}{\Leftrightarrow} \mathcal{L}(A) = \texttt{in} \\
&\overset{\text{Def. 30}}{\Leftrightarrow} \exists A' \in \mathfrak{Sup}(A), \forall B \in \mathit{Att}(A'), \mathcal{L}(B) = \texttt{out} \\
&\overset{\text{Trivial}}{\Leftrightarrow} \exists \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{H}, \mathcal{L}(B) = \texttt{out} \\
&\overset{\text{Prop. 3}}{\Leftrightarrow} \exists \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{H}, \forall B' \in \mathfrak{Sup}(B), \mathcal{L}(B') = \texttt{out} \\
&\overset{\text{Def. 49}}{\Leftrightarrow} \exists \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = \texttt{out} \\
&\overset{\text{Lem. 86}}{\Leftrightarrow} \exists \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \forall \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = \texttt{out} \\
&\overset{\text{Lem. 85}}{\Leftrightarrow} \forall \mathbf{T} \in \mathrm{Tr}[\{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}], \exists \mathcal{S} \in \mathbf{T}, \mathcal{L}'(\mathcal{S}) = \texttt{out} \\
&\overset{\text{Def. 48}}{\Leftrightarrow} \forall \mathbf{T} \in \mathit{Att}_{\mathcal{B}}(\mathfrak{Sup}(A)), \exists \mathcal{S} \in \mathbf{T}, \mathcal{L}'(\mathcal{S}) = \texttt{out}.
\end{aligned}$$

$$\begin{aligned}
&\mathcal{L}'(\mathfrak{Sup}(A)) = \texttt{out} \\
&\overset{\text{Def. 49}}{\Leftrightarrow} \forall A' \in \mathfrak{Sup}(A), \mathcal{L}(A') = \texttt{out} \\
&\overset{\text{Prop. 3}}{\Leftrightarrow} \mathcal{L}(A) = \texttt{out} \\
&\overset{\text{Def. 30}}{\Leftrightarrow} \forall A' \in \mathfrak{Sup}(A), \exists B \in \mathit{Att}(A'), \mathcal{L}(B) = \texttt{in} \\
&\overset{\text{Trivial}}{\Leftrightarrow} \forall \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \mathcal{L}(B) = \texttt{in} \\
&\overset{\text{Prop. 3}}{\Leftrightarrow} \forall \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \exists B' \in \mathfrak{Sup}(B), \mathcal{L}(B') = \texttt{in} \\
&\overset{\text{Def. 49}}{\Leftrightarrow} \forall \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \mathcal{L}'(\mathfrak{Sup}(B)) = \texttt{in} \\
&\overset{\text{Lem. 86}}{\Leftrightarrow} \forall \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \exists \mathcal{S} \in \mathbf{H}', \mathcal{L}'(\mathcal{S}) = \texttt{in} \\
&\overset{\text{Lem. 85}}{\Leftrightarrow} \exists \mathbf{T} \in \mathrm{Tr}[\{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}], \forall \mathcal{S} \in \mathbf{T}, \mathcal{L}'(\mathcal{S}) = \texttt{in} \\
&\overset{\text{Def. 48}}{\Leftrightarrow} \exists \mathbf{T} \in \mathit{Att}_{\mathcal{B}}(\mathfrak{Sup}(A)), \forall \mathcal{S} \in \mathbf{T}, \mathcal{L}'(\mathcal{S}) = \texttt{in};
\end{aligned}$$

b) if $\mathcal{L}$ is a complete labeling of $\mathfrak{A}_{\mathcal{B}}$, then $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$, denoted as $\mathcal{L}'$, is a β-complete labeling of $\mathcal{B}$.

Assume $\mathcal{L}$ is complete. We will show $\mathcal{L}'$ is β-complete. For any $A \in \mathcal{A}$, the following holds:

$$\begin{aligned}
&\mathcal{L}'(A) = \texttt{in}\\
&\overset{\text{Def. }49}{\Leftrightarrow} \mathcal{L}(\mathfrak{Sup}(A)) = \texttt{in}\\
&\overset{\text{Def. }18}{\Leftrightarrow} \forall \mathbf{X} \in \mathit{Att}_{\mathcal{B}}(\mathfrak{Sup}(A)), \exists \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{out}\\
&\overset{\text{Def. }48}{\Leftrightarrow} \forall \mathbf{X} \in \mathrm{Tr}[\{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}], \exists \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{out}\\
&\overset{\text{Lem. }85}{\Leftrightarrow} \exists \mathbf{T}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \forall \mathcal{S} \in \mathbf{T}', \mathcal{L}(\mathcal{S}) = \texttt{out}\\
&\overset{\text{Lem. }86}{\Leftrightarrow} \exists \mathcal{T} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \forall B \in \mathcal{T}, \mathcal{L}(\mathfrak{Sup}(B)) = \texttt{out}\\
&\overset{\text{Trivial}}{\Leftrightarrow} \exists A' \in \mathfrak{Sup}(A), \forall B \in \mathit{Att}(A'), \mathcal{L}(\mathfrak{Sup}(B)) = \texttt{out}\\
&\overset{\text{Def. }49}{\Leftrightarrow} \exists A' \in \mathfrak{Sup}(A), \forall B \in \mathit{Att}(A'), \mathcal{L}'(B) = \texttt{out}.
\end{aligned}$$

$$\begin{aligned}
&\mathcal{L}'(A) = \texttt{out}\\
&\overset{\text{Def. }49}{\Leftrightarrow} \mathcal{L}(\mathfrak{Sup}(A)) = \texttt{out}\\
&\overset{\text{Def. }18}{\Leftrightarrow} \exists \mathbf{X} \in \mathit{Att}_{\mathcal{B}}(\mathfrak{Sup}(A)), \forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in}\\
&\overset{\text{Def. }48}{\Leftrightarrow} \exists \mathbf{X} \in \mathrm{Tr}[\{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}], \forall \mathcal{X} \in \mathbf{X}, \mathcal{L}(\mathcal{X}) = \texttt{in}\\
&\overset{\text{Lem. }85}{\Leftrightarrow} \forall \mathbf{H}' \in \{\mathfrak{S}(\mathit{Att}(A')) \mid A' \in \mathfrak{Sup}(A)\}, \exists \mathcal{S} \in \mathbf{H}', \mathcal{L}(\mathcal{S}) = \texttt{in}\\
&\overset{\text{Lem. }86}{\Leftrightarrow} \forall \mathcal{H} \in \{\mathit{Att}(A') \mid A' \in \mathfrak{Sup}(A)\}, \exists B \in \mathcal{H}, \mathcal{L}(\mathfrak{Sup}(B)) = \texttt{in}\\
&\overset{\text{Trivial}}{\Leftrightarrow} \forall A' \in \mathfrak{Sup}(A), \exists B \in \mathit{Att}(A'), \mathcal{L}(\mathfrak{Sup}(B)) = \texttt{in}\\
&\overset{\text{Def. }49}{\Leftrightarrow} \forall A' \in \mathfrak{Sup}(A), \exists B \in \mathit{Att}(A'), \mathcal{L}'(B) = \texttt{in};
\end{aligned}$$

c) if $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$, then $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$. It holds that $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ $\Rightarrow$ according to item 2 above, $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}))$ is a β-complete labeling of $\mathcal{B}$ $\overset{Thm.62}{\Rightarrow}$ $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$;

d) if $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}$, then $\mathcal{L}$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$. It holds that $\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}$ $\Rightarrow$ according to item 1 above, $\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathrm{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}))$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ $\overset{Thm.63}{\Rightarrow}$ $\mathcal{L}$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$.

□

Lemma 87. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_\mathcal{B} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_\mathcal{B})$ be its corresponding *SETAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}$. If $\mathcal{L}_2$ is β-complete, the following holds:

a) $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$ iff $\mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subseteq \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;

b) $\mathtt{in}(\mathcal{L}_1) = \mathtt{in}(\mathcal{L}_2)$ iff $\mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) = \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;

c) $\mathtt{in}(\mathcal{L}_1) \subset \mathtt{in}(\mathcal{L}_2)$ iff $\mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subset \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) we prove each direction:
   - ($\Rightarrow$) suppose $\mathtt{in}(\mathcal{L}_1) \subseteq \mathtt{in}(\mathcal{L}_2)$. If $\mathfrak{Sup}(A) \in \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1))$, by Definition 49, there exists $A' \in \mathfrak{Sup}(A)$ such that $A' \in \mathtt{in}(\mathcal{L}_1)$. By our initial assumption, there exists $A' \in \mathfrak{Sup}(A)$ such that $A' \in \mathtt{in}(\mathcal{L}_2)$. By Definition 49, $\mathfrak{Sup}(A) \in \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;
   - ($\Leftarrow$) suppose $\mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subseteq \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$. If $A \in \mathtt{in}(\mathcal{L}_1)$, by Definition 49 we obtain $\mathfrak{Sup}(A) \in \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1))$. By our initial assumption, $\mathfrak{Sup}(A) \in \mathtt{in}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$. By Definition 49, there exists $A' \in \mathfrak{Sup}(A)$ such that $A' \in \mathtt{in}(\mathcal{L}_2)$. As $\mathcal{L}_2$ is β-complete, $A \in \mathtt{in}(\mathcal{L}_2)$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Lemma 88. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_\mathcal{B} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_\mathcal{B})$ be its corresponding *SETAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathcal{B}$. If $\mathcal{L}_1$ is β-complete, the following holds:

a) $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$ iff $\mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subseteq \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;

b) $\mathtt{out}(\mathcal{L}_1) = \mathtt{out}(\mathcal{L}_2)$ iff $\mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) = \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;

c) $\mathtt{out}(\mathcal{L}_1) \subset \mathtt{out}(\mathcal{L}_2)$ iff $\mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subset \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) - ($\Rightarrow$) suppose $\mathtt{out}(\mathcal{L}_1) \subseteq \mathtt{out}(\mathcal{L}_2)$. If $\mathfrak{Sup}(A) \in \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1))$, by Definition 49, for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \in \mathtt{out}(\mathcal{L}_1)$. By our initial assumption, for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \in \mathtt{out}(\mathcal{L}_2)$. By Definition 49, $\mathfrak{Sup}(A) \in \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$;
   - ($\Leftarrow$) suppose $\mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1)) \subseteq \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$. If $A \in \mathtt{out}(\mathcal{L}_1)$, as $\mathcal{L}_1$ is β-complete, we obtain that for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \in \mathtt{out}(\mathcal{L}_1)$. By Definition 49, $\mathfrak{Sup}(A) \in \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_1))$. By our initial assumption, $\mathfrak{Sup}(A) \in \mathtt{out}(\mathrm{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}_2))$. By Definition 49, for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \in \mathtt{out}(\mathcal{L}_2)$. In particular, $A \in \mathtt{out}(\mathcal{L}_2)$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Lemma 89. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be β-complete labelings of $\mathcal{B}$. The following holds:

a) $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1)) \subseteq \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$;

b) $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1)) = \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$;

c) $\texttt{undec}(\mathcal{L}_1) \subset \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1)) \subset \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) we prove each direction:

– (⇒) suppose $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$. If $\mathfrak{Sup}(A) \in \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1))$, by Definition 49, there is $A' \in \mathfrak{Sup}(A)$ such that $A' \in \texttt{undec}(\mathcal{L}_1)$ and for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \notin \texttt{in}(\mathcal{L}_1)$. As $\mathcal{L}_1$ is β-complete, $A \in \texttt{undec}(\mathcal{L}_1)$. By our initial assumption, $A \in \texttt{undec}(\mathcal{L}_2)$. As $\mathcal{L}_2$ is β-complete, for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \notin \texttt{in}(\mathcal{L}_2)$. By Definition 49, $\mathfrak{Sup}(A) \in \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$;

– (⇐) suppose $\texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1)) \subseteq \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$. Assume that $A \in \texttt{undec}(\mathcal{L}_1)$. As $\mathcal{L}_1$ is β-complete, for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \notin \texttt{in}(\mathcal{L}_1)$. By Definition 49, $\mathfrak{Sup}(A) \in \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_1))$. By our initial assumption, it follows $\mathfrak{Sup}(A) \in \texttt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L}_2))$. By Definition 49, there exists $A' \in \mathfrak{Sup}(A)$ such that $A' \in \texttt{undec}(\mathcal{L}_2)$ and for every $A' \in \mathfrak{Sup}(A)$, it holds $A' \notin \texttt{in}(\mathcal{L}_2)$. As $\mathcal{L}_2$ is β-complete, $A \in \texttt{undec}(\mathcal{L}_2)$;

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Theorem 65. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a grounded labeling of $\mathfrak{A}_{\mathcal{B}}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a preferred labeling of $\mathfrak{A}_{\mathcal{B}}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a stable labeling of $\mathfrak{A}_{\mathcal{B}}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_{\mathcal{B}}}(\mathcal{L})$ is a semi-stable labeling of $\mathfrak{A}_{\mathcal{B}}$.

Proof. We prove each item:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ and $\mathtt{in}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}))$ is $\subseteq$-minimal among all complete labelings of $\mathfrak{A}_\mathcal{B}$ (Theorem 64 and Lemma 87) iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a grounded labeling of $\mathfrak{A}_\mathcal{B}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ and $\mathtt{in}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}))$ is $\subseteq$-maximal among all complete labelings of $\mathfrak{A}_\mathcal{B}$ (Theorem 64 and Lemma 87) iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a preferred labeling of $\mathfrak{A}_\mathcal{B}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{undec}(\mathcal{L}) = \emptyset$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ and $\mathtt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})) = \emptyset$ (Theorem 64) iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a stable labeling of $\mathfrak{A}_\mathcal{B}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}$ iff $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}$ and $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}$ iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}_\mathcal{B}$ and $\mathtt{undec}(\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L}))$ is $\subseteq$-minimal among all complete labelings of $\mathfrak{A}_\mathcal{B}$ (Theorem 64 and Lemma 89) iff $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a semi-stable labeling of $\mathfrak{A}_\mathcal{B}$.

□

Corollary 66. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_\mathcal{B} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_\mathcal{B})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a grounded labeling of $\mathfrak{A}_\mathcal{B}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-grounded labeling of $\mathcal{B}$;

b) $\mathcal{L}$ is a preferred labeling of $\mathfrak{A}_\mathcal{B}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-preferred labeling of $\mathcal{B}$;

c) $\mathcal{L}$ is a stable labeling of $\mathfrak{A}_\mathcal{B}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-stable labeling of $\mathcal{B}$;

d) $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{A}_\mathcal{B}$ iff $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ is a β-semi-stable labeling of $\mathcal{B}$.

Proof. These results come from Theorems 63 and 65, by replacing each occurrence of $\mathcal{L}$ by $\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})$ in the statement of Theorem 65 and applying $\text{B2cS}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\text{cS2B}_{\mathcal{B}\to\mathfrak{A}_\mathcal{B}}(\mathcal{L})) = \mathcal{L}$ from Theorem 63. □

Theorems and proofs from Section 5.3.

Lemma 90. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_\mathfrak{A} = (\mathcal{A}_\mathfrak{A}, \mathit{Att}_\mathfrak{A}, \mathit{Sup}_\mathfrak{A})$ be its corresponding *BAF*. For any $A \in \mathcal{A}$, there exists $\mathcal{V} \in \mathbf{V}_A$ such that $(A, \mathcal{V}) \in \mathcal{A}_\mathfrak{A}$.

Proof. Let $A \in \mathcal{A}$. By the definition of *SETAF*, $\emptyset \neq \mathcal{X} \subseteq \mathcal{A}$ for any $\mathcal{X} \in \mathit{Att}(A)$. Then, $\mathcal{A}$ satisfies $\mathcal{X} \cap \mathcal{A} \neq \emptyset$ for every $\mathcal{X} \in \mathit{Att}(A)$. As $\mathcal{A}$ is finite, there is a $\subseteq$-minimal set $\mathcal{V} \subseteq \mathcal{A}$ such that $\mathcal{X} \cap \mathcal{V} \neq \emptyset$ for every $\mathcal{X} \in \mathit{Att}(A)$. Thus, there exists $\mathcal{V} \in \mathbf{V}_A$ such that $(A, \mathcal{V}) \in \mathcal{A}_\mathfrak{A}$. □

Theorem 67. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. For any labeling $\mathcal{L}$ of $\mathfrak{A}$, it holds $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$.

Proof. Let $A \in \mathcal{A}$, $\mathcal{L}' = \text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ and $\mathcal{L}'' = \text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}')$. There are three possibilities:

a) $\mathcal{L}(A) = \texttt{in} \overset{Def.51}{\Rightarrow} \mathcal{L}'((A,\mathcal{V})) = \texttt{in}$ for every $\mathcal{V} \in \mathbf{V}_A \overset{Lem.90}{\Rightarrow}$ in particular, there is some vulnerability $\mathcal{V} \in \mathbf{V}_A$ such that $\mathcal{L}'((A,\mathcal{V})) = \texttt{in} \overset{Def.51}{\Rightarrow} \mathcal{L}''(A) = \texttt{in}$;

b) $\mathcal{L}(A) = \texttt{out} \overset{Def.51}{\Rightarrow} \mathcal{L}'((A,\mathcal{V})) = \texttt{out}$ for every $\mathcal{V} \in \mathbf{V}_A \overset{Def.51}{\Rightarrow} \mathcal{L}''(A) = \texttt{out}$;

c) $\mathcal{L}(A) = \texttt{undec} \overset{Def.51}{\Rightarrow} \mathcal{L}'((A,\mathcal{V})) = \texttt{undec}$ for every $\mathcal{V} \in \mathbf{V}_A \overset{Lem.90}{\Rightarrow} \mathcal{L}'((A,\mathcal{V})) \neq \texttt{in}$ for every $\mathcal{V} \in \mathbf{V}_A$ and there is some vulnerability $\mathcal{V} \in \mathbf{V}_A$ such that $\mathcal{L}'((A,\mathcal{V})) \neq \texttt{out} \Rightarrow \mathcal{L}''(A) \neq \texttt{in}$ and $\mathcal{L}''(A) \neq \texttt{out} \overset{Def.51}{\Rightarrow} \mathcal{L}''(A) = \texttt{undec}$.

□

Theorem 68. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. For any labeling $\mathcal{L}$ of $\mathcal{B}_{\mathfrak{A}}$ respecting $\mathfrak{Sup}$, it holds $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$.

Proof. Let $(A,\mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$, $\mathcal{L}' = \text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ and $\mathcal{L}'' = \text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}')$. As $\mathcal{L}$ respects $\mathfrak{Sup}$, $\mathcal{L}((A,\mathcal{V}')) = \mathcal{L}((A,\mathcal{V}))$ for any $\mathcal{V}' \in \mathbf{V}_A$. By Definition 51, $\mathcal{L}'(A) = \mathcal{L}((A,\mathcal{V}))$ and $\mathcal{L}''((A,\mathcal{V}')) = \mathcal{L}'(A)$ for any $\mathcal{V}' \in \mathbf{V}_A$. In particular, $\mathcal{L}''((A,\mathcal{V})) = \mathcal{L}'(A) = \mathcal{L}((A,\mathcal{V}))$. □

Theorem 69. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$;

b) $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}$.

Proof. We prove each item:

a) if $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$, then $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}) = \mathcal{L}'$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$:

– $\mathcal{L}'((A,\mathcal{V})) = \texttt{in} \Leftrightarrow \mathcal{L}(A) = \texttt{in} \Leftrightarrow \forall \mathcal{X} \in Att(A), \exists X \in \mathcal{X}, \mathcal{L}(X) = \texttt{out} \Leftrightarrow \exists \mathcal{V}' \in \mathbf{V}_A, \mathcal{V}' \subseteq \texttt{out}(\mathcal{L}) \Leftrightarrow \exists (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \forall X \in \mathcal{V}', \mathcal{L}(X) = \texttt{out} \Leftrightarrow \exists (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \forall (X,\mathcal{V}'') \in Att((A,\mathcal{V}')), \mathcal{L}(X) = \texttt{out} \Leftrightarrow \exists (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \forall (X,\mathcal{V}'') \in Att((A,\mathcal{V}')), \mathcal{L}'((X,\mathcal{V}'')) = \texttt{out}$;

– $\mathcal{L}'((A,\mathcal{V})) = \texttt{out} \Leftrightarrow \mathcal{L}(A) = \texttt{out} \Leftrightarrow \exists \mathcal{X} \in Att(A), \forall X \in \mathcal{X}, \mathcal{L}(X) = \texttt{in} \Leftrightarrow \forall \mathcal{V}' \in \mathbf{V}_A, \exists X \in \mathcal{V}', \mathcal{L}(X) = \texttt{in} \Leftrightarrow \forall (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \exists X \in \mathcal{V}', \mathcal{L}(X) = \texttt{in} \Leftrightarrow \forall (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \exists (X,\mathcal{V}'') \in Att((A,\mathcal{V}')), \mathcal{L}(X) = \texttt{in} \Leftrightarrow \forall (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \exists (X,\mathcal{V}'') \in Att((A,\mathcal{V}')), \mathcal{L}'((X,\mathcal{V}'')) = \texttt{in}$;

b) if $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$, then $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}) = \mathcal{L}'$ is a complete labeling of $\mathfrak{A}$, for any $A \in \mathcal{A}$, the following holds:
   - $\mathcal{L}'(A) = \texttt{in} \Leftrightarrow \exists \mathcal{V} \in \mathbf{V}_A, \mathcal{L}((A,\mathcal{V})) = \texttt{in} \Leftrightarrow \exists \mathcal{V} \in \mathbf{V}_A, \exists (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \forall (X,\mathcal{V}'') \in \mathit{Att}((A,\mathcal{V}')), \mathcal{L}((X,\mathcal{V}'')) = \texttt{out} \Leftrightarrow \exists \mathcal{V} \in \mathbf{V}_A, \exists \mathcal{V}' \in \mathbf{V}_A, \forall X \in \mathcal{V}', \mathcal{L}'(X) = \texttt{out} \Leftrightarrow \exists \mathcal{V}' \in \mathbf{V}_A, \forall X \in \mathcal{V}', \mathcal{L}'(X) = \texttt{out} \Leftrightarrow \forall \mathcal{X} \in \mathit{Att}(A), \exists X \in \mathcal{X}, \mathcal{L}'(X) = \texttt{out}$;
   - $\mathcal{L}'(A) = \texttt{out} \Leftrightarrow \forall \mathcal{V} \in \mathbf{V}_A, \mathcal{L}((A,\mathcal{V})) = \texttt{out} \Leftrightarrow \forall \mathcal{V} \in \mathbf{V}_A, \forall (A,\mathcal{V}') \in \mathfrak{Sup}((A,\mathcal{V})), \exists (X,\mathcal{V}'') \in \mathit{Att}((A,\mathcal{V}')), \mathcal{L}((X,\mathcal{V}'')) = \texttt{in} \Leftrightarrow \forall \mathcal{V}' \in \mathbf{V}_A, \exists (X,\mathcal{V}'') \in \mathit{Att}((A,\mathcal{V}')), \mathcal{L}((X,\mathcal{V}'')) = \texttt{in} \Leftrightarrow \forall \mathcal{V}' \in \mathbf{V}_A, \exists (X,\mathcal{V}'') \in \mathit{Att}((A,\mathcal{V}')), \exists \mathcal{V}''' \in \mathbf{V}_X, \mathcal{L}((X,\mathcal{V}''')) = \texttt{in} \Leftrightarrow \forall \mathcal{V}' \in \mathbf{V}_A, \exists X \in \mathcal{V}', \exists \mathcal{V}''' \in \mathbf{V}_X, \mathcal{L}((X,\mathcal{V}''')) = \texttt{in} \Leftrightarrow \forall \mathcal{V}' \in \mathbf{V}_A, \exists X \in \mathcal{V}', \mathcal{L}'(X) = \texttt{in} \Leftrightarrow \exists \mathcal{X} \in \mathit{Att}(A), \forall X \in \mathcal{X}, \mathcal{L}'(X) = \texttt{in}$;

c) if $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$, then $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$: It holds that $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ $\Rightarrow$ according to item 2 above, $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is a complete labeling of $\mathfrak{A} \overset{Thm.67}{\Rightarrow} \mathcal{L}$ is a complete labeling of $\mathfrak{A}$;

d) if $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A}$, then $\mathcal{L}$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$: It holds that $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a complete labeling of $\mathfrak{A} \Rightarrow$ according to item 1 above, $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}} \overset{Thm.68}{\Rightarrow} \mathcal{L}$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$.

□

Lemma 91. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathfrak{A}$. It holds

a) $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$ iff $\texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subseteq \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$;

b) $\texttt{in}(\mathcal{L}_1) = \texttt{in}(\mathcal{L}_2)$ iff $\texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) = \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$;

c) $\texttt{in}(\mathcal{L}_1) \subset \texttt{in}(\mathcal{L}_2)$ iff $\texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subset \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) – ($\Rightarrow$) suppose $\texttt{in}(\mathcal{L}_1) \subseteq \texttt{in}(\mathcal{L}_2)$. If $(A,\mathcal{V}) \in \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1))$, by Definition 51, $A \in \texttt{in}(\mathcal{L}_1)$. By our initial assumption, $A \in \texttt{in}(\mathcal{L}_2)$. By Definition 51, $(A,\mathcal{V}) \in \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$.
   – ($\Leftarrow$) suppose $\texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subseteq \texttt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$. If $A \in \texttt{in}(\mathcal{L}_1)$, by Lemma 90, there exists $\mathcal{V} \in \mathbf{V}_A$ such that $(A,\mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$. By Definition 51,

$(A, \mathcal{V}) \in \texttt{in}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1))$. By our initial assumption, it follows that $(A, \mathcal{V}) \in \texttt{in}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$. By Definition 51, $A \in \texttt{in}(\mathcal{L}_2)$.

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Lemma 92. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be labelings of $\mathfrak{A}$. It holds

a) $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subseteq \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$;

b) $\texttt{undec}(\mathcal{L}_1) = \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) = \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$;

c) $\texttt{undec}(\mathcal{L}_1) \subset \texttt{undec}(\mathcal{L}_2)$ iff $\texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subset \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$.

Proof. We prove each item:

a) – ($\Rightarrow$) suppose $\texttt{undec}(\mathcal{L}_1) \subseteq \texttt{undec}(\mathcal{L}_2)$. If $(A, \mathcal{V}) \in \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1))$, by Definition 51, $A \in \texttt{undec}(\mathcal{L}_1)$. By our initial assumption, $A \in \texttt{undec}(\mathcal{L}_2)$. By Definition 51, $(A, \mathcal{V}) \in \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$.

– ($\Leftarrow$) suppose $\texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1)) \subseteq \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$. Assume that $A \in \texttt{undec}(\mathcal{L}_1)$, by Lemma 90, there exists $\mathcal{V} \in \mathbf{V}_A$ such that $(A, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$. By Definition 51, $(A, \mathcal{V}) \in \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_1))$. By our initial assumption, we have that $(A, \mathcal{V}) \in \texttt{undec}(\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}_2))$. By Definition 51, $A \in \texttt{undec}(\mathcal{L}_2)$.

b) it follows directly from point (a);

c) it follows directly from points (a) and (b).

□

Theorem 70. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. The following holds:

a) $\mathcal{L}$ is a grounded labeling of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-grounded labeling of $\mathcal{B}_{\mathfrak{A}}$;

b) $\mathcal{L}$ is a preferred labeling of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-preferred labeling of $\mathcal{B}_{\mathfrak{A}}$;

c) $\mathcal{L}$ is a stable labeling of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-stable labeling of $\mathcal{B}_{\mathfrak{A}}$;

d) $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-semi-stable labeling of $\mathcal{B}_{\mathfrak{A}}$.

Proof. We prove each item:

a) $\mathcal{L}$ is a grounded labeling of $\mathfrak{A}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ and $\texttt{in}(\mathcal{L})$ is $\subseteq$-minimal among all complete labelings of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\rightarrow\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$

and $\mathtt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}_{\mathfrak{A}}$ (Theorem 69 and Lemma 91) iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-grounded labeling of $\mathcal{B}_{\mathfrak{A}}$;

b) $\mathcal{L}$ is a preferred labeling of $\mathfrak{A}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ and $\mathtt{in}(\mathcal{L})$ is $\subseteq$-maximal among all complete labelings of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ and $\mathtt{in}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is $\subseteq$-maximal among all β-complete labelings of $\mathcal{B}_{\mathfrak{A}}$ (Theorem 69 and Lemma 91) iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-preferred labeling of $\mathcal{B}_{\mathfrak{A}}$;

c) $\mathcal{L}$ is a stable labeling of $\mathfrak{A}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ and $\mathtt{undec}(\mathcal{L}) = \emptyset$ iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ and $\mathtt{undec}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \emptyset$ (Theorem 69) iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-stable labeling of $\mathcal{B}_{\mathfrak{A}}$;

d) $\mathcal{L}$ is a semi-stable labeling of $\mathfrak{A}$ iff $\mathcal{L}$ is a complete labeling of $\mathfrak{A}$ and $\mathtt{undec}(\mathcal{L})$ is $\subseteq$-minimal among all complete labelings of $\mathfrak{A}$ iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-complete labeling of $\mathcal{B}_{\mathfrak{A}}$ and $\mathtt{undec}(\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L}))$ is $\subseteq$-minimal among all β-complete labelings of $\mathcal{B}_{\mathfrak{A}}$ (Theorem 69 and Lemma 92) iff $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a β-semi-stable labeling of $\mathcal{B}_{\mathfrak{A}}$.

□

Corollary 71. Let $\mathcal{B} = (\mathcal{A}, \mathit{Att}, \mathit{Sup})$ be a *BAF* and $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), \mathit{Att}_{\mathcal{B}})$ be its corresponding *SETAF*. The following holds:

a) $\mathcal{L}$ is a β-grounded labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a grounded labeling of $\mathfrak{A}$;

b) $\mathcal{L}$ is a β-preferred labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a preferred labeling of $\mathfrak{A}$;

c) $\mathcal{L}$ is a β-stable labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a stable labeling of $\mathfrak{A}$;

d) $\mathcal{L}$ is a β-semi-stable labeling of $\mathcal{B}_{\mathfrak{A}}$ iff $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ is a semi-stable labeling of $\mathfrak{A}$.

Proof. These results come from Theorems 68 and 70, by replacing each occurrence of $\mathcal{L}$ by $\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})$ in the statement of Theorem 70 and applying $\text{S2cB}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\text{cB2S}_{\mathfrak{A}\to\mathcal{B}_{\mathfrak{A}}}(\mathcal{L})) = \mathcal{L}$ from Theorem 68. □

Theorems and proofs from Section 5.4.

Proposition 72. Let $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ be its corresponding *BAF*. For any $X \in \mathcal{A}_{\mathfrak{A}}$, there is no $X' \in \mathfrak{Sup}_{\mathfrak{A}}(X)$ such that $\mathit{Att}_{\mathfrak{A}}(X') \subset \mathit{Att}_{\mathfrak{A}}(X)$.

Proof. Let $(A, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$. By reflexivity, $(A, \mathcal{V}) \in \mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}))$ and therefore $\mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}))$ is in the set $\{\mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}')) \mid (A, \mathcal{V}') \in \mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}))\}$. Now, we will show that $\mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}))$ is $\subseteq$-minimal. Let $(A, \mathcal{V}') \in \mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}))$ such that $\mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}')) \subseteq \mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}))$. This means $\mathcal{V}' \subseteq \mathcal{V}$. As $\mathcal{V}$ and $\mathcal{V}'$ are both in $V_A = \text{Tr}[\mathit{Att}(A)]$ (a set of minimal sets), we conclude $\mathcal{V} = \mathcal{V}'$. □

Proposition 73. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. It holds $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*RFBAF*.

Proof. First, we prove that $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*BAF*. We will show that *Sup* is irreflexive, symmetric, and for any $(A, B), (B, C) \in Sup$ with $A \neq C$, it holds $(A, C) \in Sup$.

a) Let $(A, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$. By Definition 50, $((X, Y), (X', Y')) \in Sup_{\mathfrak{A}}$ iff $X = X'$ and $Y \neq Y'$. Then, $((A, \mathcal{V}), (A, \mathcal{V})) \notin Sup_{\mathfrak{A}}$. We conclude $Sup_{\mathfrak{A}}$ is irreflexive.

b) Let $(A, \mathcal{V}), (A', \mathcal{V}') \in \mathcal{A}_{\mathfrak{A}}$ and assume $((A, \mathcal{V}), (A', \mathcal{V}')) \in Sup_{\mathfrak{A}}$. By Definition 50, $A = A'$ and $\mathcal{V} \neq \mathcal{V}'$. Clearly, $A' = A$ and $V' \neq V$, so we obtain $((A', \mathcal{V}'), (A, \mathcal{V})) \in Sup_{\mathfrak{A}}$. We conclude $Sup_{\mathfrak{A}}$ is symmetric.

c) Let $(A, \mathcal{V}), (A', \mathcal{V}'), (A'', \mathcal{V}'') \in \mathcal{A}_{\mathfrak{A}}$, assume $((A, \mathcal{V}), (A', \mathcal{V}')) \in Sup_{\mathfrak{A}}$ and $((A', \mathcal{V}'), (A'', \mathcal{V}'')) \in Sup_{\mathfrak{A}}$ and $(A, \mathcal{V}) \neq (A'', \mathcal{V}'')$. By Definition 50, $A = A'$ and $\mathcal{V} \neq \mathcal{V}'$ and $A' = A''$ and $\mathcal{V}' \neq \mathcal{V}''$. Hence, $A = A''$. As $(A, \mathcal{V}) \neq (A'', \mathcal{V}'')$ and $A = A''$, we conclude $\mathcal{V} \neq \mathcal{V}''$. From $A = A''$ and $\mathcal{V} \neq \mathcal{V}''$, we obtain $((A, \mathcal{V}), (A'', \mathcal{V}'')) \in Sup_{\mathfrak{A}}$.

We have shown that $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*BAF*. Now we prove that $\mathcal{B}_{\mathfrak{A}}$ is a *RFBAF*. By absurd, suppose there are arguments $(A, \mathcal{V}), (A', \mathcal{V}') \in \mathcal{A}_{\mathfrak{A}}$ such that $(A, \mathcal{V}) \neq (A', \mathcal{V}')$ and $\mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V})) = \mathfrak{Sup}_{\mathfrak{A}}((A', \mathcal{V}'))$ and $Att_{\mathfrak{A}}((A, \mathcal{V})) = Att_{\mathfrak{A}}((A', \mathcal{V}'))$. From $\mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V})) = \mathfrak{Sup}_{\mathfrak{A}}((A', \mathcal{V}'))$, we obtain $A = A'$. From $Att_{\mathfrak{A}}((A, \mathcal{V})) = Att_{\mathfrak{A}}((A', \mathcal{V}'))$, we can prove $\mathcal{V} = \mathcal{V}'$:

a) let $B \in \mathcal{V}$. By Lemma 90, argument B has some vulnerability $\mathcal{X} \in \mathbf{V}_B$ such that $(B, \mathcal{X}) \in \mathcal{A}_{\mathfrak{A}}$. By Definition 50, $(B, \mathcal{X}) \in Att_{\mathfrak{A}}((A, \mathcal{V}))$. Since $Att_{\mathfrak{A}}((A, \mathcal{V})) = Att_{\mathfrak{A}}((A', \mathcal{V}'))$, we obtain $(B, \mathcal{X}) \in Att_{\mathfrak{A}}((A', \mathcal{V}'))$. By Definition 50, $B \in \mathcal{V}'$. We proved $\mathcal{V} \subseteq \mathcal{V}'$;

b) the proof of $\mathcal{V}' \subseteq \mathcal{V}$ follows by symmetry.

As $A = A'$ and $\mathcal{V} = \mathcal{V}'$, we have $(A, \mathcal{V}) = (A', \mathcal{V}')$, a contradiction. Therefore, $\mathcal{B}_{\mathfrak{A}}$ is a *RFBAF*. As it is both a $\mathfrak{S}$-*BAF* and a *RFBAF*, we have shown $\mathcal{B}_{\mathfrak{A}}$ is a $\mathfrak{S}$-*RFBAF*. □

Proposition 74. Let $\mathfrak{A} = (\mathcal{A}, Att)$ be a *SETAF* and $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, Att_{\mathfrak{A}}, Sup_{\mathfrak{A}})$ be its corresponding *BAF*. It holds $\mathcal{B}_{\mathfrak{A}}$ is of support-guided attacks.

Proof. Let $A, B \in \mathcal{A}_{\mathfrak{A}}$ with $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$. Then, $\texttt{Conc}(A) = \texttt{Conc}(B)$. From this equality, $\texttt{Conc}(A) \in \texttt{Vul}(X) \Leftrightarrow \texttt{Conc}(B) \in \texttt{Vul}(X)$ for any $X \in \mathcal{A}_{\mathfrak{A}}$, i.e., $(A, X) \in Att_{\mathfrak{A}} \Leftrightarrow (B, X) \in Att_{\mathfrak{A}}$ for any $X \in \mathcal{A}_{\mathfrak{A}}$. □

Lemma 93. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* of support-guided attacks. For any $X, Y \in \mathcal{A}$, it holds $X \in Att(Y)$ iff $\mathfrak{Sup}(X) \in \mathfrak{S}(Att(Y))$.

Proof. Let $X, Y \in \mathcal{A}$. We prove each direction:

a) $(\Rightarrow)$ it follows directly from Definition 47;

b) $(\Leftarrow)$ assume $\mathfrak{Sup}(X) \in \mathfrak{S}(Att(Y))$. By Definition 47, there exists $X' \in Att(Y)$ such that $\mathfrak{Sup}(X') = \mathfrak{Sup}(X)$. As $\mathcal{B}$ has support-guided attacks, we have $X \in Att(Y)$.

□

Lemma 94. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a *BAF* of support-guided attacks. For any $X, Y \in \mathcal{A}$, it holds $Att(X) = Att(Y)$ iff $\mathfrak{S}(Att(X)) = \mathfrak{S}(Att(Y))$.

Proof. Let $X, Y \in \mathcal{A}$. We prove each direction:

a) $(\Rightarrow)$ trivial;

b) $(\Leftarrow)$ assume $\mathfrak{S}(Att(X)) = \mathfrak{S}(Att(Y))$. Then it follows $Z \in Att(X)$ iff (Lemma 93) $\mathfrak{Sup}(Z) \in \mathfrak{S}(Att(X))$ iff $\mathfrak{Sup}(Z) \in \mathfrak{S}(Att(Y))$ iff (Lemma 93) $Z \in Att(Y)$.

□

Lemma 95. Let $\mathcal{B} = (\mathcal{A}, Att, Sup)$ be a $\mathfrak{S}$-*BAF* of minimal and support-guided attacks. For any $A \in \mathcal{A}$, every element of $\{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$ is $\subseteq$-minimal.

Proof. Let $A \in \mathcal{A}$ and $A' \in \mathfrak{Sup}(A)$. We will show that $\mathfrak{S}(Att(A'))$ is $\subseteq$-minimal among elements of $\{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$. Assume the contrary. Thus, there exists $\mathfrak{S}(Att(B)) \subset \mathfrak{S}(Att(A'))$ for some $B \in \mathfrak{Sup}(A)$. We will prove $Att(B) \subseteq Att(A')$. Let $X \in Att(B)$. Then, $\mathfrak{Sup}(X) \in \mathfrak{S}(Att(B)) \subset \mathfrak{S}(Att(A'))$. By Lemma 93, $X \in Att(A')$. As $\mathfrak{S}(Att(B)) \neq \mathfrak{S}(Att(A'))$, we have $Att(B) \neq Att(A')$ from Lemma 94. As $Att(B) \subseteq Att(A')$ and $Att(B) \neq Att(A')$, it follows $Att(B) \subset Att(A')$. As $\mathcal{B}$ is of minimal attacks, $\mathfrak{Sup}(B) \neq \mathfrak{Sup}(A')$. As $\mathcal{B}$ is a $\mathfrak{S}$-*BAF*, from $B \in \mathfrak{Sup}(A)$ it follows $\mathfrak{Sup}(B) = \mathfrak{Sup}(A) = \mathfrak{Sup}(A')$, a contradiction. □

Theorem 75. Let $\mathcal{B}$ be a *BAF*, $\mathfrak{A}_{\mathcal{B}}$ be its corresponding *SETAF*, and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ be the corresponding *BAF* of $\mathfrak{A}_{\mathcal{B}}$. It holds $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are isomorphic iff $\mathcal{B}$ is a $\mathfrak{S}^*$-*RFBAF*.

Proof. Denote $\mathcal{B} = (\mathcal{A}, Att, Sup)$, $\mathfrak{A}_{\mathcal{B}} = (\mathfrak{S}(\mathcal{A}), Att_{\mathcal{B}})$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}} = (\mathcal{A}_{\mathfrak{A}_{\mathcal{B}}}, Att_{\mathfrak{A}_{\mathcal{B}}}, Sup_{\mathfrak{A}_{\mathcal{B}}})$. By $\mathfrak{Sup}(A)$, we mean the set of supporters of A in $\mathcal{B}$. In this proof, we will use boldface *and* italic notation (e.g., $\boldsymbol{V}$) to represent elements of $2^{2^{2^{\mathcal{A}}}}$. This only happens for $\boldsymbol{V}_{\mathfrak{Sup}(A)}$ and $\boldsymbol{V}_{\mathcal{Y}}$ (both which use a subscript), while occurrences of $\mathbf{V}$ (not italic) have no subscript, so that both cases (boldface and italic; just boldface) can be differentiated just by noticing whether there is a subscript. Besides, we explicitly declare each variable to avoid confusion. We recall that plain letters (e.g., V) represent elements of $\mathcal{A}$, caligraphic letters (e.g., $\mathcal{V}$) represents elements of $2^{\mathcal{A}}$, and boldface letters (e.g., $\mathbf{V}$) represent elements of $2^{2^{\mathcal{A}}}$. We prove each direction:

a) ($\Rightarrow$) assume $\mathcal{B}$ is not a $\mathfrak{S}^*$-*RFBAF*. By Propositions 72 and 73, $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ is a $\mathfrak{S}^*$-*RFBAF* and then it is not isomorphic to $\mathcal{B}$;

b) ($\Leftarrow$) assume $\mathcal{B}$ is a $\mathfrak{S}^*$-*RFBAF*. We will show that $\mathcal{B}$ and $\mathcal{B}_{\mathfrak{A}_{\mathcal{B}}}$ are isomorphic. Define $f : \mathcal{A} \to \mathcal{A}_{\mathfrak{A}_{\mathcal{B}}}$ such that $f(A) = (\mathfrak{Sup}(A), \mathfrak{S}(Att(A)))$.

First, we prove $\boldsymbol{V}_{\mathfrak{Sup}(A)} = \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$. By Definitions 48 and 50, $\boldsymbol{V}_{\mathfrak{Sup}(A)} = \mathrm{Tr}[Att_{\mathcal{B}}(\mathfrak{Sup}(A))] = \mathrm{Tr}[\mathrm{Tr}[\{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}]]$. By Lemma 95, $\{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$ only has minimal elements. By Theorem 1, it follows $\boldsymbol{V}_{\mathfrak{Sup}(A)} = \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$.

It remains to prove the following results for f:

– **(well-defined)** this means $f(A) \in \mathcal{A}_{\mathfrak{A}_{\mathcal{B}}}$ for any $A \in \mathcal{A}$. By Definition 50, elements of $\mathcal{A}_{\mathfrak{A}_{\mathcal{B}}}$ are pairs $(\mathcal{Y}, \mathbf{V})$ in which $\mathcal{Y} \in \mathfrak{S}(\mathcal{A})$ and $\mathbf{V} \in \boldsymbol{V}_{\mathcal{Y}}$, i.e., $\mathbf{V}$ is a vulnerability of $\mathcal{Y}$. Let $A \in \mathcal{A}$. Clearly, $\mathfrak{Sup}(A) \in \mathfrak{S}(\mathcal{A})$. As $\boldsymbol{V}_{\mathfrak{Sup}(A)} = \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$ and $A \in \mathfrak{Sup}(A)$, we obtain $\mathfrak{S}(Att(A)) \in \boldsymbol{V}_{\mathfrak{Sup}(A)}$;

– **(injective)** let $A, B \in \mathcal{A}$ and assume $f(A) = f(B)$. Then, $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and $\mathfrak{S}(Att(A)) = \mathfrak{S}(Att(B))$. By Lemma 94, $Att(A) = Att(B)$. As $\mathcal{B}$ is a *RFBAF*, $Att(A) = Att(B)$ and $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$, we conclude $A = B$;

– **(surjective)** let $(\mathfrak{Sup}(A), \mathbf{V}) \in \mathcal{A}_{\mathfrak{A}_{\mathcal{B}}}$. By Definition 50, $\mathfrak{Sup}(A) \in \mathfrak{S}(\mathcal{A})$ and $\mathbf{V} \in \boldsymbol{V}_{\mathfrak{Sup}(A)}$. As $\boldsymbol{V}_{\mathfrak{Sup}(A)} = \{\mathfrak{S}(Att(A')) \mid A' \in \mathfrak{Sup}(A)\}$, there is some $A' \in \mathfrak{Sup}(A)$ such that $\mathbf{V} = \mathfrak{S}(Att(A'))$. As $\mathcal{B}$ is a $\mathfrak{S}$-*BAF*, we have $\mathfrak{Sup}(A') = \mathfrak{Sup}(A)$. We conclude $(\mathfrak{Sup}(A), \mathbf{V}) = (\mathfrak{Sup}(A), \mathfrak{S}(Att(A'))) = (\mathfrak{Sup}(A'), \mathfrak{S}(Att(A'))) = f(A')$;

– **(f preserves *Att*)** it holds $f(A) \in Att_{\mathfrak{A}_{\mathcal{B}}}(f(B))$ iff $(\mathfrak{Sup}(A), \mathfrak{S}(Att(A)))$ is in $Att_{\mathfrak{A}_{\mathcal{B}}}((\mathfrak{Sup}(B), \mathfrak{S}(Att(B))))$ iff (Definition 50) $\mathfrak{Sup}(A) \in \mathfrak{S}(Att(B))$ iff (Lemma 93) $A \in Att(B)$;

– **(f preserves *Sup*)** it holds $f(A) \in Sup_{\mathfrak{A}_{\mathcal{B}}}(f(B))$ iff $(\mathfrak{Sup}(A), \mathfrak{S}(Att(A)))$ is in $Sup_{\mathfrak{A}_{\mathcal{B}}}((\mathfrak{Sup}(B), \mathfrak{S}(Att(B))))$ iff (Definition 50) $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and also $\mathfrak{S}(Att(A)) \neq \mathfrak{S}(Att(B))$ iff (Lemma 94) $\mathfrak{Sup}(A) = \mathfrak{Sup}(B)$ and $Att(A) \neq Att(B)$ iff ($\mathcal{B}$ is $\mathfrak{S}$-*RFBAF*) $A \in Sup(B)$.

□

Theorem 76. Let $\mathfrak{A}$ be a *SETAF*, $\mathcal{B}_{\mathfrak{A}}$ be its corresponding *BAF*, and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ be the corresponding

SETAF of $\mathcal{B}_{\mathfrak{A}}$. It holds $\mathfrak{A}$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}}$ are isomorphic.

Proof. Denote $\mathfrak{A} = (\mathcal{A}, \mathit{Att})$, $\mathcal{B}_{\mathfrak{A}} = (\mathcal{A}_{\mathfrak{A}}, \mathit{Att}_{\mathfrak{A}}, \mathit{Sup}_{\mathfrak{A}})$ and $\mathfrak{A}_{\mathcal{B}_{\mathfrak{A}}} = (\mathcal{A}_{\mathcal{B}_{\mathfrak{A}}}, \mathit{Att}_{\mathcal{B}_{\mathfrak{A}}})$. We write $\mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}))$ to mean the set of supporters of $(A, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$ in $\mathcal{B}_{\mathfrak{A}}$. Define $f : \mathcal{A} \to \mathcal{A}_{\mathcal{B}_{\mathfrak{A}}}$ such that $f(A) = \{(A, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_A\}$. It follows from Definition 50 that for any $A \in \mathcal{A}$ and $\mathcal{V}' \in \mathbf{V}_A$ we may write $f(A) = \mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}'))$.

It remains to prove the following results for f:

a) **(well-defined)** this means $f(A) \in \mathcal{A}_{\mathcal{B}_{\mathfrak{A}}}$ for any $A \in \mathcal{A}$. By Definition 48, $\mathcal{A}_{\mathcal{B}_{\mathfrak{A}}} = \{\mathfrak{Sup}_{\mathfrak{A}}((X, \mathcal{V})) \mid (X, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}\}$. Thus, we have to find $(X, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}$ such that $\mathfrak{Sup}_{\mathfrak{A}}((X, \mathcal{V})) = f(A)$. By Lemma 90, there exists $\mathcal{V}' \in \mathbf{V}_A$ such that $(A, \mathcal{V}') \in \mathcal{A}_{\mathfrak{A}}$. Then, $f(A) = \mathfrak{Sup}_{\mathfrak{A}}((A, \mathcal{V}'))$;

b) **(injective)** let $A, B \in \mathcal{A}$ and assume $f(A) = f(B)$. Then, $\{(A, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_A\} = \{(B, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_B\}$. By Lemma 90, there exists $\mathcal{V}' \in \mathbf{V}_A$ such that $(A, \mathcal{V}') \in \mathcal{A}_{\mathfrak{A}}$. Therefore, $(A, \mathcal{V}') \in \{(A, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_A\} = \{(B, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_B\}$. We conclude $(A, \mathcal{V}') \in \{(B, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_B\}$ and thus $A = B$;

c) **(surjective)** by Definitions 48 and 50, $\mathcal{A}_{\mathcal{B}_{\mathfrak{A}}} = \{\mathfrak{Sup}_{\mathfrak{A}}((X, \mathcal{V})) \mid (X, \mathcal{V}) \in \mathcal{A}_{\mathfrak{A}}\}$. Let $\mathfrak{Sup}_{\mathfrak{A}}((X, \mathcal{V})) \in \mathcal{A}_{\mathcal{B}_{\mathfrak{A}}}$. It holds $\mathfrak{Sup}_{\mathfrak{A}}((X, \mathcal{V})) = f(X)$;

d) **(f preserves *Att*)** recall that, from Definition 47, we denote $\mathfrak{S}_{\mathfrak{A}}(\mathcal{S}) = \{\mathfrak{Sup}_{\mathfrak{A}}(A) \mid$

$A \in \mathcal{S}\}$ for any $\mathcal{S} \subseteq \mathcal{A}$. For any $\mathcal{X} \subseteq \mathcal{A}$ and $A \in \mathcal{A}$, it holds

$$\{f(X) \mid X \in \mathcal{X}\} \in \mathit{Att}_{\mathcal{B}_{\mathfrak{A}}}(f(A))$$
$$\overset{\text{Def. } f}{\Leftrightarrow} \{f(X) \mid X \in \mathcal{X}\} \in \mathit{Att}_{\mathcal{B}_{\mathfrak{A}}}(\{(A, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_A\})$$
$$\overset{\text{Def. 48}}{\Leftrightarrow} \{f(X) \mid X \in \mathcal{X}\}$$
$$\in \mathrm{Tr}[\{\mathfrak{S}_{\mathfrak{A}}(\mathit{Att}_{\mathfrak{A}}((A', \mathcal{V}'))) \mid (A', \mathcal{V}') \in \{(A, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_A\}\}]$$
$$\overset{\text{Def. 50}}{\Leftrightarrow} \{f(X) \mid X \in \mathcal{X}\} \in \mathrm{Tr}[\{\mathfrak{S}_{\mathfrak{A}}(\mathit{Att}_{\mathfrak{A}}((A, \mathcal{V}))) \mid \mathcal{V} \in \mathbf{V}_A\}]$$
$$\overset{\text{Def. 50}}{\Leftrightarrow} \{f(X) \mid X \in \mathcal{X}\} \in \mathrm{Tr}[\{\mathfrak{S}_{\mathfrak{A}}(\{(X, \mathcal{V}'') \mid X \in \mathcal{V}, \mathcal{V}'' \in \mathbf{V}_X\}) \mid \mathcal{V} \in \mathbf{V}_A\}]$$
$$\overset{\text{Def. 47}}{\Leftrightarrow} \{f(X) \mid X \in \mathcal{X}\} \in \mathrm{Tr}[\{\{\{(X, \mathcal{V}'') \mid \mathcal{V}'' \in \mathbf{V}_X\} \mid X \in \mathcal{V}\} \mid \mathcal{V} \in \mathbf{V}_A\}]$$
$$\overset{\text{Def. } f}{\Leftrightarrow} \{\{(X, \mathcal{V}'') \mid \mathcal{V}'' \in \mathbf{V}_X\} \mid X \in \mathcal{X}\}$$
$$\in \mathrm{Tr}[\{\{\{(X, \mathcal{V}'') \mid \mathcal{V}'' \in \mathbf{V}_X\} \mid X \in \mathcal{V}\} \mid \mathcal{V} \in \mathbf{V}_A\}]$$
$$\overset{\text{Def. 3}}{\Leftrightarrow} \forall \mathcal{V} \in \mathbf{V}_A[\mathbf{H}_{\mathcal{V}} \cap \mathbf{T} \neq \emptyset] \text{ and } \forall \mathbf{T}' \subset \mathbf{T}, \exists \mathcal{V} \in \mathbf{V}_A[\mathbf{H}_{\mathcal{V}} \cap \mathbf{T}' = \emptyset], \text{ where}$$
$$\mathbf{H}_{\mathcal{V}} = \{\{(X, \mathcal{V}'') \mid \mathcal{V}'' \in \mathbf{V}_X\} \mid X \in \mathcal{V}\} \text{ and } \mathbf{T} = \{\{(X, \mathcal{V}) \mid \mathcal{V} \in \mathbf{V}_X\} \mid X \in \mathcal{X}\}$$
$$\Leftrightarrow \forall \mathcal{V} \in \mathbf{V}_A, \exists X \in \mathcal{V} \cap \mathcal{X} \text{ and } \forall \mathcal{X}' \subset \mathcal{X}, \exists \mathcal{V} \in \mathbf{V}_A[\mathcal{V} \cap \mathcal{X}' = \emptyset]$$
$$\overset{\text{Def. 3}}{\Leftrightarrow} \mathcal{X} \in \mathrm{Tr}[\mathbf{V}_A]$$
$$\overset{\text{Def. 50}}{\Leftrightarrow} \mathcal{X} \in \mathrm{Tr}[\mathrm{Tr}[\mathit{Att}(A)]]$$
$$\overset{\text{Thm. 1}}{\Leftrightarrow} \mathcal{X} \in \mathit{Att}(A).$$

In the last step, it holds $\mathrm{Tr}[\mathrm{Tr}[\mathit{Att}(A)]] = \mathit{Att}(A)$ by Theorem 1, as $\mathit{Att}(A)$ is a set of minimal attacks in a *SETAF* (Definition 16).

□