On the Correspondence of Non-flat Assumption-based Argumentation and Logic Programming with Negation as Failure in the Head

Anna Rapberger[1,*], Markus Ulbricht[2] and Francesca Toni[1]

[1]Imperial College London, Department of Computing
[2]Leipzig University, ScaDS.AI

Abstract
The relation between (a fragment of) assumption-based argumentation (ABA) and logic programs (LPs) under stable model semantics is well-studied. However, for obtaining this relation, the ABA framework needs to be restricted to being flat, i.e., a fragment where the (defeasible) assumptions can never be entailed, only assumed to be true or false. Here, we remove this restriction and show a correspondence between non-flat ABA and LPs with negation as failure in their head. We then extend this result to so-called set-stable ABA semantics, originally defined for the fragment of non-flat ABA called bipolar ABA. We showcase how to define set-stable semantics for LPs with negation as failure in their head and show the correspondence to set-stable ABA semantics.

Keywords
Computational Argumentation, Assumption-based Argumentation, Logic Programming, Stable Semantics

1. Introduction

Computational argumentation and logic programming constitute fundamental research areas in the field of knowledge representation and reasoning. The correspondence between both research areas has been investigated extensively, revealing that the computational argumentation and logic programming paradigms are inextricably linked and provide orthogonal views on non-monotonic reasoning. In recent years, researchers developed and studied various translations between *logic programs (LPs)* and several argumentation formalisms, including translation from and to abstract argumentation [1, 2, 3], assumption-based argumentation [4, 5, 6, 7, 8], argumentation frameworks with collective attacks [9], claim-augmented argumentation frameworks [10, 11], and abstract dialectical frameworks [12, 13].

The multitude of different translations sheds light on the close connection of negation as failure and argumentative conflicts. Apart from the theoretical insights, these translations are also practically enriching for both paradigms as they enable the application of methods developed for one of the formalisms to the other. On the one hand, translating logic programs to instances of formal argumentation has been proven useful for explaining logic programs [14]. Translations from argumentation frameworks into logic programs, on the other hand, allows to utilise the rich toolbox for LPs, e.g., answer set programming solvers like clingo [15], directly on instances of formal argumentation.

Existing translations consider *normal LPs* [16], i.e., the class of LPs in which the head of each rule amounts precisely to one positive atom. In this work, we take one step further and consider *LPs with negation as failure in the head* of rule [17]. We investigate the relation of this more general class of LPs to *assumption-based argumentation (ABA)* [4]. This is a versatile structured argumentation formalism which models argumentative reasoning on the basis of assumptions and inference rules. ABA can be suitably deployed in multi-agent settings to support dialogues [18] and supports applications in, e.g., healthcare [19], law [20] and robotics [21].

Research in ABA often focuses on the so-called *flat ABA* fragment, which prohibits deriving assumptions from inference rules. In this work, we show that generic (potentially non-flat) ABA (referred to improperly but compactly as *non-flat ABA* [22]) captures the more general fragment of LPs with negation as failure in the head, differently from all of the aforementioned argumentation formalisms. This underlines the increased and more flexible modelling capacities of the generic ABA formalism.

In this work, we investigate the relationship between non-flat ABA and LP with negation in the head, focusing on *stable* [4] and *set-stable* [23] semantics. While stable semantics is well understood, the latter has not been studied thoroughly so far. Set-stable semantics has been originally introduced for a restricted non-flat ABA fragment (*bipolar ABA* [23]) only, with the goal to study the correspondence between ABA and a generalisation of abstract argumentation that allows for support between arguments (bipolar argumentation [24]). In this paper we adopt it for any non-flat ABA framework and study it in the context of LPs with negation as failure in the head.

In more detail, our contributions are as follows:

- We show that each LP with negation as failure in the head corresponds to a non-flat ABA framework under stable semantics.
- We identify an ABA fragment (*LP-ABA*) in which the correspondence to LPs with negation as failure in the head is 1-1. We prove that each non-flat ABA framework corresponds to an LPs with negation as failure in the head by showing that each ABA framework can be mapped into an LP-ABA framework.
- We introduce set-stable model semantics for LPs with negation as failure in the head. We identify the LP fragment corresponding to bipolar ABA under set-stable semantics. We furthermore consider the set-stable semantics for any LPs with negation as failure in the head by appropriate adaptions of the reduct underpinning stable models [17].

22nd International Workshop on Nonmonotonic Reasoning, November 2-4, 2024, Hanoi, Vietnam
*Corresponding author.
a.rapberger@imperial.ac.uk (A. Rapberger);
mulbricht@informatik.uni-leipzig.de (M. Ulbricht); ft@imperial.ac.uk (F. Toni)

2. Background

We recall logic programs with negation as failure in the head [17] and assumption-based argumentation [4].

2.1. Logic programs with negation as failure in head

A *logic program with negation as failure (naf) in the head* [17] (LP in short in the remainder of the paper) consists of a set of rules r of the form

$$a_0 \leftarrow a_1, \ldots, a_m, not\ b_{m+1}, \ldots, not\ b_n$$
$$not\ a_0 \leftarrow a_1, \ldots, a_m, not\ b_{m+1}, \ldots, not\ b_n$$

for $n \geq 0$, (propositional) atoms a_i, b_i, and naf operator not. We write $head(r) = a_0$ and $head(r) = not\ a_0$, respectively, and $body(r) = \{a_1, \ldots, a_m, not\ b_{m+1}, \ldots, not\ b_n\}$. Furthermore, we let $body^+(r) = \{a_1, \ldots, a_m\}$ denote the positive and $body^-(r) = \{b_{m+1}, \ldots, b_n\}$ denote the negative atoms occuring in the body of r; moreover, we let $head^-(r) = \{a_0\}$ if $head(r) = not\ a_0$ and $head^-(r) = \emptyset$ otherwise (analogously for $head^+(r)$).

Definition 2.1. *The* Herbrand Base *of an LP P is the set HB_P of all atoms occurring in P. By*

$$\overline{HB_P} = \{not\ p \mid p \in HB_P\}$$

we denote the set of all naf-negated atoms in HB_P.

We call an LP P a *normal program* if $head^-(r) = \emptyset$ for each $r \in P$ and a *positive program* if $body^-(r) = head^-(r) = \emptyset$ for each $r \in P$. Given $I \subseteq HB_P$, the *reduct* P^I of P is the positive program

$$P^I = \{head^+(r) \leftarrow body^+(r) \mid body^-(r) \cap I = \emptyset,\ head^-(r) \subseteq I\}.$$

In contrast to the LP fragment that we consider in this work, the reduct of a program can contain (denial integrity) constraints, i.e., rules with empty head.

We are ready to define stable LP semantics.

Definition 2.2. *$I \subseteq HB_P$ is a* stable model *[17] of an LP P if I is a $\subseteq$-minimal Herbrand model of P^I, i.e., I is a $\subseteq$-minimal set of atoms satisfying*

(a) $p \in I$ iff there is a rule $r \in P^I$ s.t. $head(r) = p$ and $body(r) \subseteq I$;

(b) there is no rule $r \in P^I$ with $head(r) = \emptyset$ and $body(r) \subseteq I$.

Negation as failure in the head can be also interpreted in terms of denial integrity constraints, as also observed by Janhunen [25]. Thus, naf literals and constraints are, to some extent, two sides of the same coin. Let us consider the following example.

Example 2.3. *Consider the LP P given as follows.*

$$P:\ p \leftarrow not\ q \quad q \leftarrow not\ p \quad s \leftarrow \quad not\ s \leftarrow s, not\ p.$$

Here, P models a choice between p and q. However, as s is factual and $not\ p$ entails $not\ s$ (together with the fact s), q is rendered impossible.

For the sets of atoms $I_1 = \{p, s\}$ and $I_2 = \{q, s\}$ we obtain the following reducts:

$$P^{I_1}:\ p \leftarrow \qquad s \leftarrow$$

$$P^{I_2}:\ q \leftarrow \qquad s \leftarrow \qquad \emptyset \leftarrow s$$

We see that I_1 is a minimal Herbrand model of P^{I_1}, whereas I_2 is rendered invalid due to the rule $\emptyset \leftarrow s$. Thus, this rule can be seen as a denial integrity constraint amounting to ruling out the atom s.

2.2. Assumption-based Argumentation

We recall assumption-based argumentation (ABA) [4]. A *deductive system* is a pair $(\mathcal{L}, \mathcal{R})$, where $\mathcal{L}$ is a formal language, i.e., a set of sentences, and $\mathcal{R}$ is a set of inference rules over $\mathcal{L}$. A rule $r \in \mathcal{R}$ has the form

$$a_0 \leftarrow a_1, \ldots, a_n$$

for $n \geq 0$, with $a_i \in \mathcal{L}$. We denote the head of r by $head(r) = a_0$ and the (possibly empty) body of r with $body(r) = \{a_1, \ldots, a_n\}$.

Definition 2.4. *An* ABA framework (ABAF) *[22] is a tuple $(\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ })$ for $(\mathcal{L},\mathcal{R})$ a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ the* assumptions, *and $\overline{\ }:\mathcal{A} \to \mathcal{L}$ a* contrary *function.*

In this work, we focus on *finite* ABAFs, i.e., $\mathcal{L}$, $\mathcal{R}$, $\mathcal{A}$ are finite; also, $\mathcal{L}$ is a *set of atoms or naf-negated atoms.*

For a set of assumptions $S \subseteq \mathcal{A}$, we let $\overline{S} = \{\overline{a} \mid a \in S\}$ denote the set of all contraries of assumptions $a \in S$.

Below, we recall the fragment of *bipolar ABAFs* [23].

Definition 2.5. *An ABAF $(\mathcal{L},\mathcal{R},\mathcal{A},\overline{\ })$ is* bipolar *iff for all rules $r \in \mathcal{R}$, it holds that $|body(r)| = 1$, $body(r) \subseteq \mathcal{A}$, and $head(r) \in \mathcal{A} \cup \overline{\mathcal{A}}$.*

Next, we recall the crucial notion of *tree-derivations.* A sentence $s \in \mathcal{L}$ is *tree-derivable* from assumptions $S \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $S \vdash_R s$, if there is a finite rooted labeled tree T s.t. the root is labeled with s; the set of labels for the leaves of T is equal to S or $S \cup \{\top\}$, where $\top \notin \mathcal{L}$; for every inner node v of T there is exactly one rule $r \in R$ such that v is labelled with $head(r)$, and for each $a \in body(r)$ the node v has a distinct child labelled with a; if $body(r) = \emptyset$, v has a single child labelled $\top$; for every rule in R there is a node in T labelled by $head(r)$. We often write $S \vdash_R p$ simply as $S \vdash p$. Tree-derivations are the arguments in ABA; we use both notions interchangeably.

Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ be an ABAF. For a set of assumptions S, by $Th_D(S) = \{p \in \mathcal{L} \mid \exists S' \subseteq S : S' \vdash p\}$ we denote the set of all sentences derivable from (subsets of) S. Note that $S \subseteq Th_D(S)$ since each $a \in \mathcal{A}$ is derivable from $\{a\}$ and rule-set $\emptyset$ ($\{a\} \vdash_\emptyset a$). The *closure* of S is given by $cl(S) = Th_D(S) \cap \mathcal{A}$. An ABAF is *flat* if each set S of assumptions is closed. We refer to an ABAF not restricted to be flat as *non-flat.*

Definition 2.6. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ be an ABAF. An assumption-set $S \subseteq \mathcal{A}$* attacks *an assumption-set $T \subseteq \mathcal{A}$ if $\overline{a} \in Th_D(S)$ for some $a \in T$. An assumption-set S is* conflict-free *($S \in cf(D)$) if it does not attack itself; it is* closed *if $cl(S) = S$.*

We recall stable [22] and set-stable [23] ABA semantics (abbr. stb and sts, respectively). Note that, while set-stable semantics has been defined for bipolar ABAFs only, we generalise the semantics to arbitrary ABAFs.

Definition 2.7. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ be an ABAF. Further, let $S \in cf(D)$ be closed.*

- $S \in stb(D)$ *if* S *attacks each* $\{x\} \subseteq \mathcal{A} \setminus S$;
- $S \in sts(D)$ *if* S *attacks* $cl(\{x\})$ *for each* $x \in \mathcal{A} \setminus S$.

Example 2.8. *We consider an ABAF* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ *with assumptions* $\mathcal{A} = \{a, b, c\}$, *their contraries* $\overline{a}$, $\overline{b}$, *and* $\overline{c}$, *respectively, and rules*

$$\overline{b} \leftarrow c. \qquad b \leftarrow a. \qquad \overline{c} \leftarrow a, b.$$

The framework is non-flat *because we can derive* b *from* a.

In D, *the set* $\{c\}$ *is set-stable: Clearly, the assumption does not attack itself. It remains to show that the closure of* a *and the closure of* b *is attacked. First note that* c *attacks* b *since* $\{c\} \vdash \overline{b}$. *Thus,* c *attacks also the closure of* b. *It follows that* c *furthermore attacks the closure of* a *since* $cl(\{a\}) = \{a, b\}$. *This shows that* $\{c\}$ *is set-stable.*

Moreover, the set $\{a, b\}$ *is stable and set-stable in* D *because it is conflict-free and attacks the assumption* c *via the argument* $\{a, b\} \vdash \overline{c}$.

3. Stable Semantics Correspondence

In this section, we show that non-flat ABA under stable semantics correspond to stable model semantics for logic programs with negation as failure in the head. First, we show that each LP can be translated into a non-flat ABAF; second, we present a translation from a restricted class of ABAFs (LP-ABA) into LPs; third, we extend the correspondence result to general ABAFs by providing a translation from general non-flat ABA into LP-ABA. We conclude this section by discussing denial integrity constraints in non-flat ABA.

3.1. From LPs to ABAFs

Each LP P can be interpreted as ABAF with assumptions $not\ p$ and contraries thereof, for each literal in the Herbrand base HB_P of P. We recall the translation from normal programs to flat ABA [4].

Definition 3.1. *The* ABAF corresponding to an LP P *is* $D_P = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ *with* $\mathcal{L} = HB_P \cup \overline{HB_P}$, $\mathcal{R} = P$, $\mathcal{A} = \overline{HB_P}$, *and* $\overline{not\ x} = x$ *for each* $not\ x \in \mathcal{A}$.

Example 3.2. *Consider again the LP from Example 2.3.*

$$P:\ p \leftarrow not\ q \quad q \leftarrow not\ p \quad s \leftarrow \quad not\ s \leftarrow s, not\ p$$

Here $D_P = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ *is the ABAF with*

$$\mathcal{L} = \{p, q, s, not\ p, not\ q, not\ s\}$$
$$\mathcal{R} = P$$
$$\mathcal{A} = \{not\ p, not\ q, not\ s\}$$

and contrary function $\overline{not\ x} = x$ *for each* $x \in \{p, q, s\}$. *Recall that* $I_1 = \{p, s\}$ *is a stable model of* P. *Naturally, this set corresponds to the singleton assumption-set* $S = \{not\ q\}$. *Indeed, since* p *is derivable from* $\{not\ q\}$ *and* s *is factual, it holds that* $Th_{D_P}(S) = \{not\ q, p, s\}$ *which suffices to see that* $S \in stb(D_P)$.

Let us generalize the observations we made in this example. We translate a set of atoms I (in HB_P for an LP P) into an assumption-set $\Delta(I)$ (in the ABAF D_P) by collecting all assumptions "$not\ p$" corresponding to the atoms *outside* I; that is, we set

$$\Delta(I) = \{not\ p \mid p \notin I\}.$$

We will prove that I is a stable model (in P) iff $\Delta(I)$ is a stable extension (in D_P). First, we introduce a notion of reachability in logic programs that is based on the construction of arguments.

Definition 3.3. *Let* P *be an LP. An atom* $p \in HB_P \cup \overline{HB_P}$ *is* reachable *from a set of naf literals* $N \subseteq \overline{HB_P}$ *iff there is a tree-based argument* $N' \vdash p$ *with* $N' \subseteq N$ *in the corresponding ABAF* D_P.

Note that the reachability target is defined for both positive and negative atoms; the source on the other hand is always a set of naf literals. The notion differs from reachability based on dependency graphs which is defined for positive atoms only.

Below, we prove our first main result.

Theorem 3.4. *Let* P *be an LP and* D_P *the ABAF corresponding to* P. *Then* I *is a stable model of* P *iff* $\Delta(I) \in stb(D_P)$.

Proof. By definition, a set I is stable iff it is $\subseteq$-minimal model of P^I satisfying

(a) $p \in I$ iff there is $r \in P^I$ such that $head(r) = p$ and $body(r) \subseteq I$; and
(b) there is no $r \in P^I$ with $head(r) = \emptyset$ and $body^+(r) \subseteq I$.

By definition of P^I we obtain I is a stable model of P iff I is a $\subseteq$-minimal model of P^I satisfying

(a) $p \in I$ iff there is $r \in P$ such that $head(r) = p$, $body^+(r) \subseteq I$, and $body^-(r) \cap I = \emptyset$; and
(b) there is no $r \in P$ with $head^+(r) = \emptyset$, $head^-(r) \subseteq I$, $body^+(r) \subseteq I$, and $body^-(r) \cap I = \emptyset$.

Below, we show that the first item and the $\subseteq$-minimality requirement captures conflict-freeness (no naf literal in I is derived) and the requirement that all other assumptions are attacked (all other naf literals outside I are derived); whereas the second item ensures closure of the program.[1]

First, Let I be a stable model of P and let $S = \Delta(I)$. We show that S is stable in D_P, i.e., it is conflict-free, closed, and attacks all assumptions in $\mathcal{A} \setminus S$.

- <u>S is conflict-free:</u> S is conflict-free iff there is no $p \in HB_P \setminus I$ such that p is reachable, i.e., can be derived from S. If such a derivation would exist, then the assumption $not\ p \in S$ were attacked by S. Towards a contradiction, suppose there is an atom $p \in HB_P \setminus I$ which is reachable from S. Let

  $$Q = \{p \in HB_P \setminus I \mid S \vdash p\}$$

  denote the set of atoms that are reachable from S but lie 'outside' I. We order Q according the height of the smallest tree-derivation.

  Wlog, we can assume that our chosen atom p is minimal in Q, i.e., there is no other atom $q \in HB_P \setminus I$ which is reachable in less steps. Let $S' \vdash p$ denote the smallest tree-derivation, and let r denote the top-rule (the rule connecting the root p with the fist level of the tree) of the derivation. The rule satisfies $head(r) = p$, $body^-(r) \cap I = \emptyset$, and $body^+(r) \subseteq I$

[1] We note that in the case of normal logic programs without negation in the head, the second condition does not apply. It is well known and has been discussed thoroughly in the literature that (a) holds iff $\Delta(I)$ is stable in D_P [5, 6].

(otherwise, there is an atom $q \notin I$ with a smaller tree-derivation, contradiction to the minimality of p in Q). Consequently, we obtain that $p \in I$, contradiction to our initial assumption.

- S attacks all other assumptions: Suppose there is an atom $p \in I$ which is not reachable from S. We show that $I' = I \setminus \{p\}$ is a model of P^I. That is, we show that I' satisfies each rule in P^I. By assumption there is is no rule $r \in P$ such that $head(r) = p$, $body^+(r) \subseteq I'$, and $body^-(r) \cap I' = \emptyset$ (otherwise, p is reachable from S). Hence $p \in I'$ iff there is $r \in P^I$ such that $head(r) = p$ and $body^+(r) \subseteq I'$ is satisfied. I' satisfies all constraints since, by assumption, there is no $r \in P^I$ with $head(r) = \emptyset$ and $body^+(r) \subseteq I$. Thus I' is a model of P^I. Consequently, I cannot be a stable model, contradiction to our initial assumption.
- S is closed: Towards a contradiction, suppose that there is some $p \in I$ such that the corresponding naf literal $not\ p$ is reachable. Let r be the top-rule of the tree-derivation. It holds that $body^+(r) \subseteq I$ (otherwise, there is some $q \in HB_P \setminus I$ which is reachable, contradiction to the first item), $body^-(r) \cap I = \emptyset$ and $head(r) = not\ p$. Consequently, item (b) from Definition 2.2 is violated.

This concludes the proof of the first direction. We have shown that $S = \Delta(I)$ is stable in D_P.

Now, let $S = \Delta(I)$ be a stable extension in D_P. We show that I is stable in P.

- Let $p \in I$. Then we can construct an argument $S' \vdash p$, $S' \subseteq S$ in D_P, i.e., is reachable from S. We show that there is a rule r with $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$ and $head(r) = p$. We proceed by induction over the height of the argument, that is, the height of the tree-derivation.
  - Base case: Suppose $S' \vdash p$ has height 1. Then there is $r \in P$ with $head(r) = p$, $body^+(r) = \emptyset$, and $body^-(r) \cap S = \emptyset$.
  - $n \mapsto n + 1$: Suppose now that the statement holds for all arguments of height smaller than or equal to n, and suppose $S' \vdash p$ has height $n + 1$. Let r denote the top-rule of the tree-derivation.
    We derive the statement by applying the induction hypothesis to all height-maximal sub-arguments (with claims in $body(r)$) of our fixed tree-derivation: Let $p' \in body(r)$. The sub-tree with root p' is an argument of height n. Hence, by induction hypothesis, $\Delta(I)$ derives p', i.e., there is $r' \in P$ with $head(r') = p'$, $body^+(r') \subseteq I$, and $body^-(r') \cap I = \emptyset$. In case p' is a positive literal, we obtain $p' \in I$ (by (a) from Definition 2.2); in case p' is a naf literal, we obtain $p' \in \Delta(I)$ (by (b)). Since p' was arbitrary, we obtain $body^+(r) \subseteq I$ and $body^-(r) \cap I = \emptyset$.
- For the other direction, suppose there is a rule $r \in P$ with $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$ and $head(r) = p$. We can construct arguments for all $body^+(r) \subseteq I$ and thus obtain $p \in I$.
- Towards a contradiction, suppose there is a $r \in P$ with $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$ and $head(r) = not\ p$ for some $p \in I$. Then we can construct an argument for $not\ p$, contradiction to S being closed.
- It remains to show that I is a $\subseteq$-minimal model of P^I. Since each atom $p \in I$ has an argument in D_P we obtain minimality: Towards a contradiction, suppose there is a model $I' \subsetneq I$ of P^I. Let $p \in I \setminus I'$. Since there is an argument deriving p there is some $r \in P^I$ with $head(r) = p$ and $body(r) \subseteq I$, showing that I' is not a model of P^I. □

3.2. From ABAFs to LPs

For the other direction, we define a mapping so that each assumption corresponds to a naf-negated atom. However, we need to take into account that ABA is a more general formalism. Indeed, in LPs, there is a natural bijection between ordinary atoms and naf-negated ones (i.e., p corresponds to $not\ p$). Instead, in ABAFs, assumptions can have the same contrary, they can be the contraries of each other, and not every sentence is the contrary of an assumption in general. To show the correspondence (under stable semantics), we proceed in two steps:

1. We define the *LP-ABA* fragment in which i) no assumption is a contrary, ii) each assumption has a unique contrary, and iii) no further sentences exist, i.e., each element in $\mathcal{L}$ is either an assumption or the contrary of an assumption. We show that the translation from such LP-ABAFs to LPs is semantics-preserving.
2. We show that each ABAF (whose underpinning language is restricted to atoms and their naf) can be transformed to an LP-ABAF whilst preserving semantics.

Relating LP and LP-ABA Let us start by defining the LP-ABA fragment. A similar fragment for the case of normal LPs and flat ABAFs has been already considered [6, 22, 26]. Here, we extend it to the more general case.

Definition 3.5. *The* LP-ABA fragment *is the class of all ABAFs* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ *where (1)* $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$*, (2) the contrary function* $\overline{}$ *is injective, and (3)* $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$.

We show that each LP-ABAF corresponds to an LP, using a translation similar to [6][Definition 11] (which is however for flat ABA). We replace each assumption a with $not\ \overline{a}$. For an atom $p \in \mathcal{L}$, we let

$$rep(p) = \begin{cases} not\ \overline{p}, & \text{if } p \in \mathcal{A} \\ \overline{a}, & \text{if } p = \overline{a} \in \overline{\mathcal{A}}. \end{cases}$$

Note that in the LP-ABA fragment, this case distinction is exhaustive. We extend the operator to ABA rules element-wise: $rep(r) = rep(head(r)) \leftarrow \{rep(p) \mid r \in body(r)\}$.

Definition 3.6. *For an LP-ABAF* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$*, we define the* associated LP $P_D = \{rep(r) \mid r \in \mathcal{R}\}$.

Example 3.7. *Let D be an ABAF with $\mathcal{A} = \{p, q, s\}$ and*

$$\mathcal{R} : \overline{p} \leftarrow q \qquad \overline{q} \leftarrow p \qquad \overline{s} \leftarrow \qquad s \leftarrow \overline{s}, p.$$

We replace e.g. the assumption p with $not\ \overline{p}$ and the contrary $\overline{p}$ is left untouched. This yields the associated LP

$$P_D : \overline{p} \leftarrow not\ \overline{q} \quad \overline{q} \leftarrow not\ \overline{p} \quad \overline{s} \leftarrow \quad not\ \overline{s} \leftarrow \overline{s}, not\ \overline{p}.$$

Striving to anticipate the relation between D and P_D, note that $S = \{q\} \in stb(D)$. Now we compute $Th_D(S) \setminus \mathcal{A} = \{\overline{p}, \overline{s}\}$ noting that it is a stable model of P_D.

It can be shown that, when restricting to LP-ABA, the translations in Definitions 3.1 and 3.6 are each other's inverse. Below, we let

$$rep(D) = (rep(\mathcal{L}), rep(\mathcal{R}), rep(\mathcal{A}), \overline{})$$

where $\overline{rep(a)} = \overline{a}$.

Lemma 3.8. *For any LP P, it holds that $P = P_{D_P}$.*

Proof. Each naf atom $not\ p$ corresponds to an assumption in P_D whose contrary is p. Applying the translation from Definition 3.6, we map each assumption $not\ p$ to the naf literal $not\ \overline{not\ p} = not\ p$. Hence, we reconstruct the original LP P. □

We obtain a similar result for the other direction, under the assumption that each literal is the contrary of an assumption, i.e., if $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$ as it is the case for the LP-ABA fragment. The translations from Definition 3.6 and 3.1 are each other's inverse modulo the simple assumption renaming operator rep as defined above. Note that we associate each assumption $a \in \mathcal{A}$ with $not\ \overline{a}$.

Lemma 3.9. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF in the LP fragment. It holds that $D_{P_D} = rep(D)$.*

Proof. When applying the translation from ABA to LP ABA, we associate each assumption $a \in \mathcal{A}$ with a naf literal $not\ \overline{a}$. Applying the translation from Definition 3.1, each naf literal $not\ \overline{a}$ is an assumption in D_{P_D}. We obtain $D_{P_D} = (rep(\mathcal{L}), rep(\mathcal{R}), rep(\mathcal{A}), \overline{})$ where $\overline{rep(a)} = \overline{a}$. □

We are ready to prove the main result of this section. We make use of Theorem 3.4 and obtain the following result.

Theorem 3.10. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an LP-ABAF and let P_D be the associated LP. Then, $S \in stb(D)$ iff $Th_D(S) \setminus \mathcal{A}$ is a stable model of P_D.*

Proof. It holds that S is stable in D iff

$$rep(S) = \{not\ \overline{a} \mid a \in S\}$$

is stable in $rep(D)$. This in turn is equivalent to $rep(S)$ is stable in D_{P_D} (by Proposition 3.9). Equivalently,

$$\{\overline{a} \mid not\ \overline{a} \notin rep(S)\} = \{\overline{a} \mid a \notin S\} = Th_D(S) \setminus \mathcal{A}$$

is stable in P_D (by Proposition 3.4). This in turn holds iff $Th_D(S) \setminus \mathcal{A}$ is stable in P_D (by definition, $P_D = \{rep(r) \mid r \in \mathcal{R}\} = P_D$). □

From ABA to LP-ABA To complete the correspondence result between ABA and LP, it remains to show that each ABAF D can be mapped to an LP-ABAF D'. To do so, we proceed as follows:

1. For each assumption $a \in \mathcal{A}$ we introduce a fresh atom c_a; in the novel ABAF D', c_a is the contrary of a.
2. If p is the contrary of a in the original ABAF D, then we add a rule $c_a \leftarrow p$ to D'.
3. For any atom p that is neither an assumption nor a contrary in D, we add a fresh assumption a_p and let p be the contrary of a_p.

Example 3.11. *Consider the ABAF D with literals $\mathcal{L} = \{a, b, c, p, q\}$, assumptions $\mathcal{A} = \{a, b, c\}$, and their contraries $\overline{a} = p$, $\overline{b} = p$, and $\overline{c} = a$, respectively, with rules*

$$\mathcal{R}: \ r_1 = p \leftarrow a, b \quad r_2 = q \leftarrow a, b \quad r_3 = p \leftarrow c.$$

First note that $\{c\} \in stb(D)$. We construct the LP-ABAF D' by adding rules $c_a \leftarrow p$, $c_b \leftarrow p$, and $c_c \leftarrow a$; c_a, c_b, and c_c are the novel contraries. Moreover, q is neither a contrary nor an assumption, so we add a novel assumption a_q with contrary q. The stable extension $\{c\}$ is only preserved under projection: we now have $\{c, a_q\} \in stb(D')$.

We show that each ABAF D can be mapped into an (under projection) equivalent LP-ABAF D'. We furthermore note that the translation can be computed efficiently.

Proposition 3.12. *For each ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ there is ABAF D' computable in polynomial time s.t. (i) D' is an LP-ABAF and (ii) $S \in stb(D')$ iff $S \cap \mathcal{A} \in stb(D)$.*

Proof. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF and let $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}, \overline{}')$ be ABAF constructed as described, i.e.,

1. For each assumption $a \in \mathcal{A}$ we introduce a fresh atom c_a; in the novel ABAF D', c_a is the contrary of a.
2. If p is the contrary of a in the original ABAF D, then we add a rule $c_a \leftarrow p$.
3. For any atom p that is neither an assumption nor a contrary in D, we add a fresh assumption a_p and let p be the contrary of a_p in D'.

First of all, the construction is polynomial. Towards the semantics, let us denote the result of applying steps (1) and (2) by D^*. We show that in D and D^* the attack relation between semantics persists.

Let $S \subseteq \mathcal{A}$ be a set of assumptions. In the following, we make implicit use of the fact that entailment in D and D^* coincide except the additional rules deriving certain contraries in D^*.

($\Rightarrow$) Suppose S attacks a in D for some $a \in \mathcal{A}$. Then $p \in Th_D(S)$ where $p = \overline{a}$. By construction, $p \in Th_{D^*}(S)$ as well and since $p = \overline{a}$, the additional rule $c_a \leftarrow p$ is applicable. Consequently, $c_a \in Th_{D^*}(S)$, i.e., S attacks a in D^* as well.

($\Leftarrow$) Now suppose S attacks a in D^* for some $a \in \mathcal{A}$. Then $c_a \in Th_{D^*}(S)$ which is only possible whenever $p \in Th_{D^*}(S)$ holds for p the original contrary of a. Thus S attacks a in D.

We deduce

$$stb(D) = stb(D^*).$$

Finally, for moving from D^* to D' we note that adding assumptions a_p (which do not occur in any rule) corresponds to adding arguments without outgoing attacks to the constructed AF F_{D^*}. This has (under projection) no influence on the stable extensions of D^*. Consequently

$$S \in stb(D') \ \Leftrightarrow\ S \cap \mathcal{A} \in stb(D^*) \ \Leftrightarrow\ S \cap \mathcal{A} \in stb(D).$$

as desired. □

Given an ABAF D, we combine the previous translation with Definition 3.6 to obtain the associated LP P_D. Thus, each ABAF D can be translated into an LP, as desired.

Example 3.13. *Let us consider again the ABAF D from Example 3.11. As outlined before, applying the translation into an LP-ABA D' yields an ABAF D' with assumptions $\mathcal{A} = \{a, b, c, a_q, a_p\}$ their contraries $\overline{a} = c_a$, $\overline{b} = c_b$, $\overline{c} = c_c$, $\overline{a_q} = q$, and $\overline{a_p} = p$, respectively, and with rules*

$$p \leftarrow a, b. \qquad q \leftarrow a, b. \qquad p \leftarrow c.$$
$$c_a \leftarrow p. \qquad c_b \leftarrow p. \qquad c_c \leftarrow a.$$

The resulting framework lies in the LP-ABA class. In the next step, we apply the translation from LP-ABA to LP and obtain the associated LP P_D with rules

$$p \leftarrow not\ c_a, not\ c_b. \quad q \leftarrow not\ c_a, not\ c_b. \quad p \leftarrow not\ c_c.$$
$$c_a \leftarrow p. \qquad c_b \leftarrow p. \quad c_c \leftarrow not\ c_a.$$

The set $\{p, c_a, c_b\}$ is the stable model corresponding to our stable extension $\{c\}$ from D (under projection).

3.3. Denial Integrity Constraints in ABA

Our correspondence results allow for a novel interpretation of the derivation of assumptions in ABA in the context of stable semantics. Analogous to the correspondence of naf in the head and allowing for constraints (rules with empty head) in LP we can view the derivation of an assumption as *setting constraints*: for a set of assumptions $M \subseteq \mathcal{A}$ and an assumption $a \in \mathcal{A}$, a derivation $M \vdash a$ intuitively captures the constraint $\leftarrow M, \overline{a}$, i.e., one of $M \cup \{\overline{a}\}$ is false.

Thus, our results indicate that deriving assumptions is the same as imposing constraints. More formally, the following observation can be made.

Proposition 3.14. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ be an ABAF and let $D' = (\mathcal{L}, \mathcal{R} \cup \{r\}, \mathcal{A}, \overline{\ })$ for a rule r of the form $a \leftarrow M$ with $M \cup \{a\} \subseteq \mathcal{A}$. Then, $S \in stb(D')$ iff (i) $S \in stb(D)$ and (ii) $M \not\subseteq S$ or $a \in S$.*

Proof. We first make the following observation. We have

$$\forall S \subseteq \mathcal{A} : Th_D(S) \subseteq Th_{D'}(S)$$

by definition and

$$p \in Th_{D'}(S) \setminus Th_D(S) \Rightarrow a \notin S$$

because the only additional way to make deriviations in D' is through a rule entailing a. This, however, implies

$$S \text{ closed in } D' \Rightarrow Th_D(S) = Th_{D'}(S), \qquad (1)$$

i.e., for sets closed in D', the derived atoms coincide.

Now let us show the equivalence.

($\Rightarrow$) Suppose $S \in stb(D')$. Since S is closed, $M \not\subseteq S$ or $a \notin S$, so condition (ii) is met. Moreover, by (1), S is conflict-free and attacks each $a \notin S$ in D, i.e., $D \in stb(D)$. Thus condition (i) is also met.

($\Leftarrow$) Let $S \in stb(D)$ and let $M \not\subseteq S$ or $a \in S$. Then S is also closed in D'. We apply (1) and find $S \in stb(D')$. $\square$

Example 3.15. *Consider the ABAF D with assumptions $\mathcal{A} = \{a, b, c, d\}$, and their contraries $\overline{a}$, $\overline{b}$, $\overline{c}$, and $\overline{d}$, respectively, with rules*

$$r_1 = \overline{c} \leftarrow a, b. \qquad r_2 = \overline{a} \leftarrow c.$$

The ABAF D has two stable models: $S_1 = \{a, b, d\}$ and $S_2 = \{b, d, c\}$.

Consider the ABAF D' where we add a new rule

$$r_3 = a \leftarrow d.$$

Intuitively, this rule encodes the constraint $\leftarrow \overline{a}, d$, *i.e., $\overline{a}$ and d cannot be true both at the same time. Consequently, the ABAF D' has a single stable model S_1.*

4. Set-Stable Model Semantics

In this section, we investigate set-stable semantics in the context of logic programs.

Set-stable semantics has been originally introduced for bipolar ABAFs (where each rule is of the form $p \leftarrow a$ with a an assumption and p either an assumption or the contrary thereof) for capturing existing notions of stable extensions for bipolar (abstract) argumentation; we will thus first identify the corresponding LP fragment of *bipolar LPs* and introduce the novel semantics therefor. We then show that this semantics corresponds to set-stable ABA semantics, even in the general case. Interestingly, despite being the formally correct counter-part to set-stable ABA semantics, the novel LP semantics exhibits non-intuitive behavior in the general case, as we will discuss.

4.1. Bipolar LPs and Set-Stable Semantics

Recall that an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ is *bipolar* iff each rule is of the form $p \leftarrow a$ where a is an assumption and p is either an assumption or the contrary of an assumption. We adapt this to LPs as follows.

Definition 4.1. *The* bipolar LP fragment *is the class of LPs P with $|body(r)| = 1$ and $body(r) \subseteq \overline{HB_P}$ for all $r \in P$.*

We note that the head of a rule corresponds by definition either to an assumption (if it is a naf literal) or the contrary of an assumption (if it is a positive literal).

We set out to define our new semantics. In ABA, set-stable semantics relaxes stable semantics: it suffices if the *closure* of an assumption a outside a given set is attacked; that is, it suffices if a "supports" an attacked assumption b, e.g., if the ABAF contains the rule $b \leftarrow a$. Let us discuss this for bipolar LPs: given a set of atoms $I \subseteq HB_P$ in a program P, we can accept an atom p not only if it is reachable from $\Delta(I)$, but also if there is some reachable q and $not\ p$ "supports" $not\ q$. For instance, given the rule of the form $not\ q \leftarrow not\ p \in P$, we are allowed to add the *contraposition* $p \leftarrow q$ to the program P before evaluating our potential model I.

To capture all "supports" between naf-negated atoms, we define their *closure*, amounting to the set of all positive and naf-negated atoms obtainable by forward chaining.

Definition 4.2. *For a bipolar LP P and a set $S \subseteq HB_P \cup \overline{HB_P}$, we define*

$$supp(S) = S \cup \{l \mid \exists r \in P : body(r) \subseteq S, head(r) = l\}.$$

The closure *of S is defined as $cl(S) = \bigcup_{i>0} supp^i(S)$.*[2]

Note that $cl(S)$ returns positive as well as negative atoms. For a singleton $\{a\}$, we write $cl(a)$ instead of $cl(\{a\})$.

Example 4.3. *Consider the bipolar LP P given as follows.*

$$P : \ p \leftarrow not\ p \qquad not\ q \leftarrow not\ p \qquad q \leftarrow not\ s.$$

Then, $cl(\{not\ p\}) = \{p, not\ q, not\ p\}$, $cl(\{not\ q\}) = \{not\ q\}$, and $cl(\{not\ s\}) = \{q, not\ s\}$.

We define a modified reduct by adding rules to make the closure explicit: for each atom $a \in HB_P$, if $not\ b$ can be reached from $not\ a$, we add the rule $a \leftarrow b$.

[2] $supp^i(S)$ denotes the i-th application of $supp(\cdot)$ to S.

Definition 4.4. *For a bipolar LP P and $I \subseteq HB_P$, the* set-stable reduct *P_s^I of P is defined as $P_s^I = P^I \cup P_s$ where*

$$P_s = \{a \leftarrow b \mid a, b \in HB_P, a \neq b, not\ b \in cl(\{not\ a\})\}.$$

Note that we require $a \neq b$ to avoid constructing redundant rules of the form "$a \leftarrow a$".

Example 4.5. *Let us consider again the LP P from Example 4.3. Let $I_1 = \{q\}$ and $I_2 = \{p, q\}$. We compute the set-stable reducts according to Definition 4.4. First, we compute the reducts P^{I_1} and P^{I_2}. Second, for each naf literal $not\ x$, we add a rule $x \leftarrow y$, for each $y \in HB_P$ with $not\ y \in cl(\{not\ x\})$, to both reducts. Inspecting the computed closures of the naf literals of P, this amounts to adding the rule $(p \leftarrow q)$ to each reduct.*

Overall, we obtain

$$\begin{array}{lllll} P_s^{I_1}: & p \leftarrow & \emptyset \leftarrow & q \leftarrow & p \leftarrow q \\ P_s^{I_2}: & & & q \leftarrow & p \leftarrow q \end{array}$$

We are ready to give the definition of set-stable semantics. Note that we state the definition for arbitrary (not only bipolar) LPs.

Definition 4.6. *An interpretation $I \subseteq HB_P$ is a* set-stable model *of an LP P if I is a $\subseteq$-minimal model of P_s^I satisfying*

(a) $p \in I$ iff there is $r \in P_s^I$ s.t. $head(r) = p$ and $body(r) \subseteq I$;

(b) there is no rule $r \in P_s^I$ with $head(r) = \emptyset$ and $body(r) \subseteq I$.

Example 4.7. *Consider again the LP P from Example 4.3. It can be checked that P has no stable model. Indeed, the reduct P^{I_1} contains the unsatisfiable rule $(\emptyset \leftarrow)$; the set $I_2 = \{p, q\}$ on the other hand is not minimal for P^{I_2}.*

If we consider the generalised set-stable reduct instead, we find that the set I_2 is a $\subseteq$-minimal model for $P_s^{I_2}$. The atom q is factual in $P_s^{I_2}$ and the atom p is derived by q. Thus, I_2 is set-stable in P.

4.2. Set-stable Semantics in general (non-bipolar) LPs

So far, we considered set-stable model semantics in the bipolar LP fragment. As it is the case for the set-stable ABA semantics, our definition of set-stable LP semantics generalises to arbitrary LPs, beyond the bipolar class.

Set-stable model semantics belong to the class of two-valued semantics, that is, each atom is either set to true or false (no undefined atoms exist). Moreover, set-stable model semantics generalises stable model semantics: each stable model of an LP is set-stable, but not vice versa, as Example 4.7 shows.

Proposition 4.8. *Let P be an LP. Each stable model I of P is set-stable (but not vice versa).*

Proof. Let I denote a stable model of P. By definition, the generalised reduct P_s^I of P^I is a superset of all rules in P^I. Thus (a) and (b) in Definition 4.6 are satisfied. Moreover, I is $\subseteq$-minimal by Definition 2.2. □

We furthermore note that the support of a set of positive and negative atoms can be computed in polynomial time.

Lemma 4.9. *For a bipolar LP P and a set $S \subseteq HB_P \cup \overline{HB_P}$, $cl(S)$ is computable in polynomial time.*

It follows that the computation of a set-stable model of a given program P is of the same complexity as finding a stable model.

In the case of general LPs, however, the novel semantics exhibits counter-intuitive behavior, as the following example demonstrates.

Example 4.10. *Consider the following two LPs P_1 and P_2:*

$$\begin{array}{lll} P_1: & q \leftarrow & not\ q \leftarrow not\ p \\ P_2: & q \leftarrow & not\ q \leftarrow not\ p, not\ s. \end{array}$$

In P_1 the set $\{p, q\}$ is set-stable because we can take the contraposition of the rule and obtain $p \leftarrow q$. This is, however, not possible in P_2 which in fact has no set-stable model.

The example indicates that the semantics does not generalise well to arbitrary LPs. We note that a possible and arguably intuitive generalisation of set-stable model semantics would be to allow for contraposition for all rules that derive a naf literal. This, however, requires disjunction in the head of rules. Applying this idea to Example 4.10 yields the rule $p \vee s \leftarrow q$ when constructing the reduct with respect to P_2. The resulting instance therefore lies in the class of disjunctive LPs (a thorough investigation of this proposal however is beyond the scope of the present paper).

4.3. Relating ABA and LP under set-stable semantics

In the previous subsection, we identified certain shortcomings of set-stable semantics when applied to general LPs. This poses the question whether our formulation of set-stable LP semantics is indeed the LP-counterpart of set-stable ABA semantics. In this subsection, we show that, despite the unwanted behavior of set-stable model semantics for LPs, the choice of our definitions is correct: set-stable ABA and LP semantics correspond to each other. We show that our novel LP semantics indeed captures the spirit of ABA set-stable semantics, even in the general case.

We show that the semantics correspondence is preserved under the translation presented in Definition 3.1. We prove the following theorem.

Theorem 4.11. *For an LP P and its associated ABAF D_P, I is set-stable in P iff $\Delta(I)$ is set-stable in D_P.*

Proof. By definition, I is set-stable iff it is a $\subseteq$-minimal model of P_s^I satisfying

(a) $p \in I$ iff there is $r \in P_s^I$ s.t. $head(r) = p$ and $body(r) \subseteq I$;

(b) there is no $r \in P_s^I$ with $head(r) = \emptyset$ and $body(r) \subseteq I$.

Equivalently, by definition of P_s^I,

(a) $p \in I$ iff

(1) there is $r \in P$ s.t. $head(r) = p$, $body^+(r) \subseteq I$ and $body^-(r) = \emptyset$; or

(2) there is $q \in I$ such that $not\ q \in cl(not\ p)$ and there is $r \in P$ s.t. $head(r) = q$, $body^+(r) \subseteq I$ and $body^-(r) = \emptyset$; and

(b) there is no $r \in P$ with $head^+(r) = \emptyset$, $head^-(r) \subseteq I$, $I \subseteq body^+(r)$, and $body^-(r) = \emptyset$.

The second item (b) is analogous to the proof of Theorem 3.4; item (a1) corresponds to item (a) of the proof of Theorem 3.4. Item (a2) formalises that it suffices to (in terms of ABA) attack the closure of a set.

Let I be a set-stable model of P. We show that $S = \Delta(I)$ is set-stable in D_P, i.e., S is conflict-free, closed, and attacks the closure of all remaining assumptions. The first two points are analogous to the proof of Theorem 3.4. Below we prove the last item.

- S attacks the closure of all other assumptions: Suppose there is an atom $p \in I$ which is not reachable from S and there is no $q \in I$ with $notq \in cl(notp)$. Similar to the proof in Theorem 3.4, we can show that $I' = I \setminus \{p\}$ is a model of P_s^I. By assumption there is is no rule $r \in P$ such that $head(r) = p$, $body^+(r) \subseteq I'$, and $body^-(r) \cap I' = \emptyset$ (otherwise, p is reachable from S); moreover, there is no rule $p \leftarrow q$ in P_s (otherwise, $not\ p$ is in the support from $not\ q$). We obtain that I' is a model of P_s^I, contradiction to our initial assumption.

Next, we prove the other direction. Let $S = \Delta(I)$ be a set-stable extension of D_P. We show that I is set-stable in P. Similar to the proof of Theorem 3.4 we can show that all constraints are satisfied and that I is indeed minimal. Also, the remaining correspondence proceeds similar as in the case of stable semantics, as shown below.

- Let $p \in I$. Then either we can construct an argument $S' \vdash p$, $S' \subseteq S$ in D_P, or there is some $q \in I$ such that $not\ q \in cl(not\ p)$ for which we can construct an argument in D_P. If the former holds, then we proceed analogously to the corresponding part in the proof of Theorem 3.4 and item (a1) is satisfied. Now, suppose the latter is true. Analogously to the the proof of Theorem 3.4, we can show that there is a rule $r \in P$ with $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$ and $head(r) = q$, that is (a2) is satisfied.
- For the other direction, suppose there is a rule $r \in P$ with $body^+(r) \subseteq I$, $body^-(r) \cap I = \emptyset$ and $head(r) = p$ and there is $q \in I$ with $not\ q \in cl(not\ p)$ and $head(r) = q$, $body^+(r) \subseteq I$ and $body^-(r) = \emptyset$ for some r. We can construct arguments for all $body^+(r) \subseteq I$ and thus $p \in I$. □

Analogous to the case of stable semantics, we can show that the LP-ABA fragment preserves the set-stable semantics and obtain the following result.

Theorem 4.12. *Let D be an LP-ABAF and let P_D be the associated LP. Then, $S \in sts(D)$ iff $Th_D(S) \setminus \mathcal{A}$ is a set-stable model of P_D.*

Making use of the translation from general ABA to the LP-ABA fragment outlined in the previous section, we obtain that the correspondence extends to general ABA.

4.4. Set-stable Semantics for General (non-bipolar) ABAFs

Recall that Example 4.10 indicates that the semantics does not generalise well in the context of LPs. In light of the close relation between ABA and LP, it might be the case that the non-intuitive behavior affects set-stable ABA semantics. However, we find that set-stable semantics generalise well for ABAFs. The reason lies in the differences between deriving assumptions (in ABA) and naf literals (in LPs) beyond classical stable model semantics.

Let us translate Example 4.10 in the language of ABA.

Example 4.13. *The translation of the LPs P_1 and P_2 from Example 4.10 yields two ABAFs D_1 and D_2. The ABAF D_1 has two assumptions $\mathcal{A}_1 = \{a, b\}$ (representing not p and not q, respectively) with contraries $\overline{a}$ and $\overline{b}$, and rules*

$$\mathcal{R}_1 : \ \overline{b} \leftarrow . \qquad\qquad b \leftarrow a.$$

The ABAF D_2 has three assumptions $\mathcal{A}_2 = \{a, b, c\}$ (representing not p, not q, and not s, respectively) with contraries $\overline{a}, \overline{b}, \overline{c}$, and rules

$$\mathcal{R}_2 : \ \overline{b} \leftarrow . \qquad\qquad b \leftarrow a, c.$$

By Theorem 4.11, we obtain the set-stable extensions of the ABAFs from our results from the original programs P_1 and P_2. In D_1, the empty set is set-stable because it attacks the closure of each assumption. In D_2, on the other hand, no set of assumptions is set-stable: a and c are not attacked, although they jointly derive b which is attacked by the empty set.

In contrast to the LP formulation of the problem where taking the contraposition of each rule with a naf literal in the head would have been a more natural solution, the application of set-stable semantics in the reformulation of Example 4.10 confirms our intuition. The set $\{a, c\}$ derives the assumption b, however, the attack onto b is not propagated to (the closure of) one of the members of $\{a, c\}$.

The example indicates a fundamental difference between deriving assumptions and naf literals in ABA and LPs, respectively. A rule in an LP with a naf literal in the head is interpreted as denial integrity constraint (under stable model semantics). As a consequence, the naf literal in the head of a rule is replaceable with any positive atom in the body; e.g., the rules $not\ p \leftarrow q, s$ and $not\ s \leftarrow q, p$ are equivalent as they both formalise the constraint $\leftarrow p, q, s$. Although a similar behavior of rules with assumptions in the head can be identified in the context of stable semantics in ABA, the derivation of an assumption goes beyond that; it indicates a hierarchical dependency between assumptions.

5. Discussion

In this work, we investigated the close relation between non-flat ABA and LPs with negation as failure in the head, focusing on stable and set-stable semantics. Research often focuses on the flat ABA fragment in which each set of assumptions is closed. This restriction has however certain limitations; as the present work demonstrates, non-flat ABA is capable of capturing a more general LP fragment, thus opening up more broader application opportunities. To the best of our knowledge, our work provides the first correspondence result between an argumentation formalism and a fragment of logic programs which is strictly larger than the class of normal LPs. We furthermore studied set-stable semantics, originally defined only for bipolar ABAFs, in context of general non-flat ABAFs and LPs.

The provided translations have practical as well as theoretical benefits. Conceptually, switching views between deriving assumptions (as possible in non-flat ABA) and imposing denial integrity constraints (as possible in many standardly considered LP fragments) allows us to look at a problem from different angles; oftentimes, it can be helpful to change viewpoints for finding solutions. Practically, our translations yield mutual benefits for both fields. Our translations from ABA into LP yield a solver for non-flat ABA instances (as, for instance, employed in [27]), as commonly used ASP solvers (like clingo [15]) can handle constraints. With this, we provide a powerful alternative to solvers for non-flat ABA, which are typically not supported by established ABA solvers due to the primary focus on flat instances (with some exceptions [28, 29]). LPs can profit from the thoroughly investigated explanation methods for ABAFs [22, 30, 31].

The generalisation of set-stable model semantics to the non-bipolar ABA and LP fragment furthermore indicated interesting avenues for future research. As Example 4.10 indicates, the semantics does not generalise well beyond the bipolar LP fragment. It would be interesting to further investigate reasonable generalisations for set-stable model semantics for LPs. As discussed previously, a promising generalisation might lead us into the fragment of disjunctive LPs. Another promising direction for future work would be to further study and develop denial integrity constraints in the context of ABA, beyond stable semantics. A further interesting avenue for future work is the development and investigation of three-valued semantics (such as partial-stable or L-stable model semantics) for LPs with negation as failure in the head, in particular in correspondence to their anticipated ABA counter-parts (e.g., complete and semi-stable semantics, respectively).

As the case of set-stable semantics indicates, it is unlikely that the correspondence between denial integrity constraints and assumptions in the head is satisfied beyond stable semantics. It would be interesting to investigate denial integrity constraints in the realm of ABA, to shed light on the relation (and differences) between the derivation of assumptions and setting constraints.

Acknowledgments

This research was funded in whole, or in part, by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101020934, ADIX) and by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme; by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

References

[1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–358.

[2] Y. Wu, M. Caminada, D. M. Gabbay, Complete extensions in argumentation coincide with 3-valued stable models in logic programming, Studia Logica 93 (2009) 383–403. doi:10.1007/s11225-009-9210-5.

[3] M. Caminada, S. Sá, J. Alcântara, W. Dvořák, On the equivalence between logic programming semantics and argumentation semantics, Int. J. Approx. Reasoning 58 (2015) 87–111. doi:10.1016/j.ijar.2014.12.004.

[4] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artif. Intell. 93 (1997) 63–101.

[5] C. Schulz, F. Toni, Logic programming in assumption-based argumentation revisited - semantics and graphical representation, in: B. Bonet, S. Koenig (Eds.), Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 1569–1575. URL: https://doi.org/10.1609/aaai.v29i1.9417. doi:10.1609/AAAI.V29I1.9417.

[6] M. Caminada, C. Schulz, On the equivalence between assumption-based argumentation and logic programming, J. Artif. Intell. Res. 60 (2017) 779–825.

[7] S. Sá, J. F. L. Alcântara, Assumption-based argumentation is logic programming with projection, in: J. Vejnarová, N. Wilson (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings, volume 12897 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 173–186. URL: https://doi.org/10.1007/978-3-030-86772-0_13. doi:10.1007/978-3-030-86772-0_13.

[8] F. Toni, N. Potyka, M. Ulbricht, P. Totis, Understanding problog as probabilistic argumentation, in: E. Pontelli, S. Costantini, C. Dodaro, S. A. Gaggl, R. Calegari, A. S. d'Avila Garcez, F. Fabiano, A. Mileo, A. Russo, F. Toni (Eds.), Proceedings 39th International Conference on Logic Programming, ICLP 2023, Imperial College London, UK, 9th July 2023 - 15th July 2023, volume 385 of *EPTCS*, 2023, pp. 183–189. URL: https://doi.org/10.4204/EPTCS.385.18. doi:10.4204/EPTCS.385.18.

[9] M. König, A. Rapberger, M. Ulbricht, Just a matter of perspective, in: F. Toni, S. Polberg, R. Booth, M. Caminada, H. Kido (Eds.), Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, UK, 14-16 September 2022, volume 353 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2022, pp. 212–223. URL: https://doi.org/10.3233/FAIA220154. doi:10.3233/FAIA220154.

[10] A. Rapberger, Defining argumentation semantics under a claim-centric view, in: S. Rudolph, G. Marreiros (Eds.), Proceedings of the 9th European Starting AI Researchers' Symposium 2020 co-located with 24th European Conference on Artificial Intelligence (ECAI 2020), Santiago Compostela, Spain, August, 2020, volume 2655 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: https://ceur-ws.org/Vol-2655/paper2.pdf.

[11] W. Dvorák, A. Rapberger, S. Woltran, A claim-centric perspective on abstract argumentation semantics: Claim-defeat, principles, and expressiveness, Artif. Intell. 324 (2023) 104011. URL: https://doi.org/10.1016/j.artint.2023.104011. doi:10.1016/J.ARTINT.2023.104011.

[12] J. F. L. Alcântara, S. Sá, J. C. A. Guadarrama, On the equivalence between abstract dialectical frameworks and logic programs, Theory Pract. Log. Program. 19

(2019) 941–956. doi:10.1017/S1471068419000280.
[13] H. Strass, Approximating operators and semantics for abstract dialectical frameworks, Artif. Intell. 205 (2013) 39–70. doi:10.1016/j.artint.2013.09.004.
[14] C. Schulz, F. Toni, Justifying answer sets using argumentation, Theory Pract. Log. Program. 16 (2016) 59–110. URL: https://doi.org/10.1017/S1471068414000702. doi:10.1017/S1471068414000702.
[15] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, Theory Pract. Log. Program. 19 (2019) 27–82. URL: https://doi.org/10.1017/S1471068418000054. doi:10.1017/S1471068418000054.
[16] T. Janhunen, Representing normal programs with clauses, in: Proceedings of ECAI 2004, IOS Press, 2004, pp. 358–362.
[17] K. Inoue, C. Sakama, Negation as failure in the head, The Journal of Logic Programming 35 (1998) 39–78. URL: https://www.sciencedirect.com/science/article/pii/S0743106697100012. doi:https://doi.org/10.1016/S0743-1066(97)10001-2.
[18] X. Fan, F. Toni, A general framework for sound assumption-based argumentation dialogues, Artif. Intell. 216 (2014) 20–54. doi:10.1016/j.artint.2014.06.001.
[19] K. Čyras, T. Oliveira, A. Karamlou, F. Toni, Assumption-based argumentation with preferences and goals for patient-centric reasoning with interacting clinical guidelines, Argument Comput. 12 (2021) 149–189.
[20] P. M. Dung, P. M. Thang, N. D. Hung, Modular argumentation for modelling legal doctrines of performance relief, Argument Comput. 1 (2010) 47–69. doi:10.1080/19462160903564584.
[21] X. Fan, S. Liu, H. Zhang, C. Leung, C. Miao, Explained activity recognition with computational assumption-based argumentation, in: Proc. ECAI, volume 285 of *FAIA*, IOS Press, 2016, pp. 1590–1591. doi:10.3233/978-1-61499-672-9-1590.
[22] K. Čyras, X. Fan, C. Schulz, F. Toni, Assumption-based argumentation: Disputes, explanations, preferences, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018, pp. 365–408.
[23] K. Cyras, C. Schulz, F. Toni, Capturing bipolar argumentation in non-flat assumption-based argumentation, in: B. An, A. L. C. Bazzan, J. Leite, S. Villata, L. W. N. van der Torre (Eds.), PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings, volume 10621 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 386–402. URL: https://doi.org/10.1007/978-3-319-69131-2_23. doi:10.1007/978-3-319-69131-2_23.
[24] C. Cayrol, M. Lagasquie-Schiex, On the acceptability of arguments in bipolar argumentation frameworks, in: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU 2005, Barcelona, Spain, July 6-8, 2005, Proceedings, volume 3571 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 378–389. URL: https://doi.org/10.1007/11518655_33. doi:10.1007/11518655_33.
[25] T. Janhunen, On the effect of default negation on the expressiveness of disjunctive rules, in: T. Eiter, W. Faber, M. Truszczynski (Eds.), Logic Programming and Nonmonotonic Reasoning, 6th International Conference, LPNMR 2001, Vienna, Austria, September 17-19, 2001, Proceedings, volume 2173 of *Lecture Notes in Computer Science*, Springer, 2001, pp. 93–106. URL: https://doi.org/10.1007/3-540-45402-0_7. doi:10.1007/3-540-45402-0_7.
[26] T. Lehtonen, J. P. Wallner, M. Järvisalo, Declarative algorithms and complexity results for assumption-based argumentation, J. Artif. Intell. Res. 71 (2021) 265–318. URL: https://doi.org/10.1613/jair.1.12479. doi:10.1613/JAIR.1.12479.
[27] F. Russo, A. Rapberger, F. Toni, Argumentative causal discovery, CoRR abs/2405.11250 (2024). URL: https://doi.org/10.48550/arXiv.2405.11250. doi:10.48550/ARXIV.2405.11250. arXiv:2405.11250.
[28] T. Lehtonen, A. Rapberger, F. Toni, M. Ulbricht, J. P. Wallner, Instantiations and computational aspects of non-flat assumption-based argumentation, CoRR abs/2404.11431 (2024). URL: https://doi.org/10.48550/arXiv.2404.11431. doi:10.48550/ARXIV.2404.11431. arXiv:2404.11431.
[29] M. Thimm, The formal argumentation libraries of tweety, in: E. Black, S. Modgil, N. Oren (Eds.), Theory and Applications of Formal Argumentation - 4th International Workshop, TAFA 2017, Melbourne, VIC, Australia, August 19-20, 2017, Revised Selected Papers, volume 10757 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 137–142. URL: https://doi.org/10.1007/978-3-319-75553-3_9. doi:10.1007/978-3-319-75553-3_9.
[30] X. Fan, S. Liu, H. Zhang, C. Miao, C. Leung, Two forms of explanations in computational assumption-based argumentation, in: K. Larson, M. Winikoff, S. Das, E. H. Durfee (Eds.), Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017, ACM, 2017, pp. 1532–1534. URL: http://dl.acm.org/citation.cfm?id=3091352.
[31] X. Fan, F. Toni, Assumption-based argumentation dialogues, in: T. Walsh (Ed.), IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011, IJCAI/AAAI, 2011, pp. 198–203. URL: https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-044. doi:10.5591/978-1-57735-516-8/IJCAI11-044.