Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200506

Argumentative Reflections of Approximation Fixpoint Theory

Jesse HEYNINCK [a,1] and Ofer ARIELI [b,2]
[a] *Department of Computer Science, TU Dortmund, Germany*
[b] *School of Computer Science, The Academic College of Tel-Aviv, Israel*

Abstract. In this paper we show that non-monotonic formalisms that are represented by approximation fixpoint theory can also be represented by formal argumentation frameworks. By this, we are able not only to recapture and generalize many forms of non-monotonic reasoning in the context of argumentation theory, but also introduce new argumentative representations that have not been considered so far.

Keywords. Non-monotonic reasoning, approximation fixpoint theory, assumption-based argumentation.

1. Introduction

In a series of papers (e.g., [7,8]) Denecker, Marek, and Truszczyński introduced a general technique, using approximating fixpoint computations, for constructively characterizing a variety of non-monotonic formalism. In [17] this method was further applied to a variety of logic programs (including normal logic programs, first order logic programs, and logic programming with aggregates), and in [1] it has been applied to HEX programs. In this paper, we show how fixpoints of approximating operators can be represented by (extensions of) the 'reflecting' assumption-based argumentation framework, thus allowing for argumentative counterparts of corresponding characterizations that were provided in terms of approximation fixpoint theory. These alternative argumentative characterizations generalize known characterizations of semantics of non-monotonic formalisms such as default logic, logic programming and autoepistemic logic that are introduced in, e.g., [3,5], and allow for new argumentative representations of other formalisms for non-monotonic reasoning.

The paper is organized as follows: in the next section we recall some basic notions from assumption-based argumentation, approximation fixpoint theory, and semantics for logic programing. In Section 3 we show how argumentation theory can be used for characterizing semantics of propositional logic programs and how this can be generalized to reflections of approximated fixpoint concepts. In Section 4 we give some applications of our results, and in Section 5 we conclude.

[1] Corresponding Author. Email: jesse.heyninck@tu-dortmund.de. Supported by the German National Science Foundation, DFG-project KE-1413/11-1.
[2] Supported by the Israel Science Foundation, grant No. 550/19.

2. Preliminaries

2.1. Assumption-Based Argumentation (ABA)

Definition 1. A *reasoning frame* for a propositional language $\mathcal{L}$ is a pair $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$, where $\vdash$ is a monotonic binary relation between sets of formulas and formulas in $\mathcal{L}$ (so if $\Gamma \vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \psi$).

The next definition, which is a variation of that in [13], generalizes the definition in [3] of assumption-based frameworks.

Definition 2. An *assumption-based framework* (ABF, for short) is a triple $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$, where:

- $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ is a reasoning frame,
- Λ (the *defeasible assumptions*) is a non-empty, countable set of $\mathcal{L}$-formulas,
- $- : \Lambda \to \wp(\mathcal{L})$ is a contrariness operator, assigning a finite set of $\mathcal{L}$-formulas to every defeasible assumption in Λ.[3]

Remark 1. In [3,13] ABFs are in fact quadruples, where the assumptions are divided to strict and defeasible ones. In what follows strict assumptions will not be needed, so they are removed from the definition. Yet, our results can be easily adjusted to ABFs with a set Γ of strict premises, defined e.g. by $\Gamma = \{\psi \mid \emptyset \vdash \psi\}$.

Remark 2. In this paper we shall concentrate on *flat* ABFs. In such ABFs the sets of assumptions are always closed, i.e., they contain any assumption they imply. Non-flat characterizations of approximate fixpoint theory will be investigated in future work.

Attacks in ABFs of assertions by counter-assertions are defined as follows:

Definition 3. Let $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$ be an assumption-based framework, $\Delta, \Theta \subseteq \Lambda$, and $\psi \in \Lambda$. We say that Δ *attacks* ψ iff $\Delta \vdash \phi$ for some $\phi \in -\psi$. Accordingly, Δ attacks Θ if Δ attacks some $\psi \in \Theta$.

The last definition gives rise to the following adaptation to ABFs of the usual semantics for abstract argumentation frameworks [9].

Definition 4. [3] Given $\mathsf{ABF} = \langle \mathfrak{L}, \Lambda, - \rangle$, we denote $\mathcal{AF}(\mathsf{ABF}) = (\wp(\Lambda), \rightsquigarrow)$ where $(\Delta, \Theta) \in \rightsquigarrow$ for some $\Delta, \Theta \subseteq \Lambda$ iff Δ attacks Θ. We denote $\Delta^+ = \{\phi \in \Lambda \mid \Delta \rightsquigarrow \phi\}$.
For $\Delta \subseteq \Lambda$, we say that

Δ is *conflict-free* iff there is no $\Delta' \subseteq \Delta$ that attacks some $\psi \in \Delta$. Δ *defends* a set $\Delta' \subseteq \Lambda$ iff for every set Θ that attacks Δ' there is a set $\Delta'' \subseteq \Delta$ that attacks Θ. Δ is *admissible* iff it is conflict-free and defends every $\Delta' \subseteq \Delta$. Δ is *complete* iff it is admissible and contains every $\Delta' \subseteq \Lambda$ that it defends. Δ is *grounded* iff it is minimally complete.[4] Δ is *preferred* iff it is maximally complete.[5] Δ is *stable* iff it is conflict-free and $\Delta^+ = \Lambda \setminus \Delta$.

[3] Here and in what follows $\wp(\mathcal{L})$ denotes the powerset of $\mathcal{L}$.

[4] For flat ABFs the grounded extension always exists and it is unique.

[5] Often, preferred extensions are defined as maximally admissible. However, for flat ABFs, these definitions are equivalent.

Example 1. Let $\mathsf{ABF}_{\mathcal{P}} = \langle \mathfrak{L}_{\mathsf{MP}}, \{\neg p, \neg q, \neg r\}, -\rangle$ be an assumption-based argumentation framework in which $-\neg x = \{x\}$ for any $x \in \{p,q,r\}$, and where $\mathfrak{L}_{\mathsf{MP}}$ is a reasoning setting whose entailment relation is based on Modus Ponens as its single rule, i.e.: $\Delta \vdash \psi$ if there is a derivation of ψ based on the formulas in Δ and the inference rule

$$[\mathsf{MP}_{\mathcal{P}}] \quad \frac{\phi_1 \quad \phi_2 \quad \cdots \quad \phi_n \quad \psi \leftarrow \phi_1, \ldots, \phi_n \in \mathcal{P}}{\psi}$$

where in this case we take: $\mathcal{P} = \{q \leftarrow \neg p;\ p \leftarrow \neg q;\ r \leftarrow \neg q;\ r \leftarrow \neg r\}$. Figure 1 below is a schematic representation of a fragment of the attack relation in ABF.

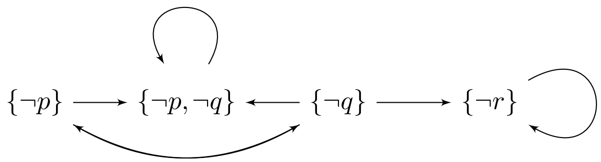

Figure 1. The ABF of Example 1

The sets $\emptyset, \{\neg p\}, \{\neg q\}$ are admissible (and complete) in ABF. The latter two are also preferred, and $\{\neg q\}$ is also stable. The grounded extension here is $\emptyset$.

2.2. Approximation Fixpoint Theory (AFT)

Next, we review the basics notions of approximation fixpoint theory (AFT, [7]). Its main purpose is to find constructive techniques for approximating the fixpoints of an operator O over a lattice L. For this, the following structure (known as a *bilattice*, see [11,12]) is useful:

Definition 5. *Given a lattice $L = \langle \mathcal{L}, \leq \rangle$, we let $L^2 = \langle \mathcal{L}^2, \leq_i, \leq_t \rangle$ be a structure in which $\mathcal{L}^2 = \mathcal{L} \times \mathcal{L}$, and for every $x_1, y_1, x_2, y_2 \in \mathcal{L}$,*

- *$(x_1, y_1) \leq_i (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \geq y_2$,*
- *$(x_1, y_1) \leq_t (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \leq y_2$.*

An approximation operator $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ of $O_{\mathcal{L}} : \mathcal{L} \to \mathcal{L}$ is defined by specifying two operators $\mathcal{O}_l$ and $\mathcal{O}_u$ which calculate a *lower* and an *upper bound* for the value of $O_{\mathcal{L}}$. It is observed in [7] that many formalisms can be characterized by a symmetric operator where the upper bound can be calculated by "inversing" the lower bound (and vice versa). This is formalized next.

Definition 6. Let $O_{\mathcal{L}} : \mathcal{L} \to \mathcal{L}$ and $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$.

- $\mathcal{O}$ is an *approximation* of $O_{\mathcal{L}}$, if $\forall x, y, \in \mathcal{L}$, $\mathcal{O}(x,y) = (\mathcal{O}_l(x,y), \mathcal{O}_u(x,y))$, where $\mathcal{O}_l : \mathcal{L}^2 \to \mathcal{L}$ and $\mathcal{O}_u : \mathcal{L}^2 \to \mathcal{L}$ are a lower and upper bound, respectively, of $O_{\mathcal{L}}(x)$ and $O_{\mathcal{L}}(y)$, namely: $\mathcal{O}_l(x,y) \leq O_{\mathcal{L}}(x)$ and $\mathcal{O}_u(x,y) \geq O_{\mathcal{L}}(y)$.

- $\mathcal{O}$ is *symmetric* if $\mathcal{O}(x,y) = (\mathcal{O}_l(x,y), \mathcal{O}_l(y,x))$ for some $\mathcal{O}_l : \mathcal{L}^2 \to \mathcal{L}$; $\mathcal{O}$ is $\leq_i$-*monotone*, if whenever $(x_1,y_1) \leq_i (x_2,y_2)$, also $\mathcal{O}(x_1,y_1) \leq_i \mathcal{O}(x_2,y_2)$; and $\mathcal{O}$ is *approximating*, if it is both symmetric and $\leq_i$-monotone.

In [7] it is shown that the *stable operator*, as defined next, can be used for expressing the semantics of many non-monotonic formalisms.

Definition 7. Let $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ be an approximation operator.

- $\mathcal{O}_l(\cdot,y) = \lambda x.\mathcal{O}_l(x,y)$, i.e., for $x \in \mathcal{L}$, $\mathcal{O}_l(\cdot,y)(x) = \mathcal{O}_l(x,y)$.
- $C(\mathcal{O}) : \mathcal{L} \to \mathcal{L}$, *the complete stable operator for* $\mathcal{O}$, is defined, for every $y \in \mathcal{L}$, by: $C(\mathcal{O})(y) = lfp(\mathcal{O}_l(.,y)) = \min_{\leq}\{x \in \mathcal{L} \mid x = \mathcal{O}_l(x,y)\}$.
- $S(\mathcal{O}) : \mathcal{L}^2 \to \mathcal{L}^2$, *the stable operator for* $\mathcal{O}$, is $S(\mathcal{O})(x,y) = (C(\mathcal{O})(y), C(\mathcal{O})(x))$.

Stable operators capture the idea of minimizing truth in the sense that for any $\leq_i$-monotone operator $\mathcal{O}$ on $\mathcal{L}^2$, fixpoints of the stable operator $S(\mathcal{O})$ are $\leq_t$-minimal fixpoints of $\mathcal{O}$ (see [7, Theorem 4]).

Accordingly, the following notions are defined in [7]:

Kripke-Kleene fixpoint of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid (x,y) = lfp_{\leq_i}(\mathcal{O}(x,y))\}$
well-founded semantics of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid (x,y) = lfp_{\leq_i}(S(\mathcal{O})(x,y))\}$
three-valued stable models of $\mathcal{O}$:	$\{(x,y) \in \mathcal{L}^2 \mid S(\mathcal{O})(x,y) = (x,y)\}$
two-valued stable models of $\mathcal{O}$:	$\{(x,x) \in \mathcal{L}^2 \mid S(\mathcal{O})(x,x) = (x,x)\}$

These semantical notions have been shown to provide a uniform framework for the mechanisms underlying many major knowledge representation formalisms, such as logic programming [17], autoepistemic logic [8], default logic [8], abstract argumentation [27] and abstract dialectical frameworks [27]. In more detail, for autoepistemic and default logics, the lattice of possible-world structures is investigated, and operators for autoepistemic logic and default logic give rise to characterizations of various formalisms, including autoepistemic expansions [16], well-founded semantics for default logic [2], weak default extensions [15], and Reiter's default extensions [20]. Full details can be found in [8].

2.3. Propositional Logic Programming (LP)

We now review some notions from logic programming theory that are needed in what follows. For simplicity and due to lack of space, we restrict our attentions to the propositional case. Following [17,25], a *generalized logic program* $\mathcal{P}$ is a finite set of the rules of the form $p \leftarrow \psi$, where p is an atom and ψ is a formula. A rule is *normal* if ψ is a conjunction of literals (that is, a conjunction of atomic formulas or negated atoms). A program is normal if it consists only of normal rules.

Given a four-valued lattice $\mathsf{F} \leq_t \mathsf{U}, \mathsf{B} \leq_t \mathsf{T}$ and a $\leq_t$-involution $-$ on it (i.e, $-\mathsf{F} = \mathsf{T}$, $-\mathsf{T} = \mathsf{F}$, $-\mathsf{U} = \mathsf{U}$ and $-\mathsf{B} = \mathsf{B}$), a four-valued *interpretation* of a generalized program $\mathcal{P}$ is a pair (x,y), where x is the set of the atoms that are assigned a value in $\{\mathsf{T},\mathsf{B}\}$ and y is the set of atoms assigned a value in $\{\mathsf{T},\mathsf{U}\}$. An interpretation (x,y) is *consistent* if $x \subseteq y$ (i.e., it doesn't have B-assignments). Truth assignments to complex formulas are then recursively defined as follows:

- $(x,y)(\phi) = \begin{cases} \mathsf{T} & \text{if } \phi \in x \text{ and } \phi \in y \\ \mathsf{U} & \text{if } \phi \notin x \text{ and } \phi \in y \\ \mathsf{F} & \text{if } \phi \notin x \text{ and } \phi \notin y \\ \mathsf{B} & \text{if } \phi \in x \text{ and } \phi \notin y \end{cases}$
- $(x,y)(\neg\phi) = -(x,y)(\phi)$
- $(x,y)(\psi \wedge \phi) = \min_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$
- $(x,y)(\psi \vee \phi) = \max_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$

The *immediate consequence operator* $\Phi_{\mathcal{P}}$ of $\mathcal{P}$ is now defined as follows:

$$\Phi_{\mathcal{P}}(x,y) = (\Phi^l_{\mathcal{P}}(x,y), \Phi^u_{\mathcal{P}}(x,y))$$

- $\Phi^l_{\mathcal{P}}(x,y) = \{\phi \in \mathsf{Atoms} \mid \text{ there is some } \phi \leftarrow \psi \in \mathcal{P}, (x,y)(\psi) \in \{\mathsf{T},\mathsf{B}\}\}$,
- $\Phi^u_{\mathcal{P}}(x,y) = \{\phi \in \mathsf{Atoms} \mid \text{ there is some } \phi \leftarrow \psi \in \mathcal{P}, (x,y)(\psi) \in \{\mathsf{T},\mathsf{U}\}\}$.

Remark 3. It can be easily seen that equivalently, one can define the immediate consequence operator $\Phi_{\mathcal{P}}$ by: $\Phi_{\mathcal{P}}(x,y) = (x',y')$, where for any atom ϕ,

$$(x',y')(\phi) = \max_{\leq_t}\{(x,y)(\psi) \mid \phi \leftarrow \psi \in \mathcal{P}\}.$$

We furthermore note that, alternatively, $\Phi^u_{\mathcal{P}}(x,y)$ can be taken as $\Phi^l_{\mathcal{P}}(y,x)$.

Note that $\Phi_{\mathcal{P}}$ is an operator on the lattice of the four-valued interpretations of $\mathcal{P}$. We therefore can define the following semantics for $\mathcal{P}$ in terms of the fixpoint notions considered in the previous section:

Definition 8. Given a generalized program $\mathcal{P}$, we say that a consistent interpretation (x,y) is:

- a *partial stable model* of $\mathcal{P}$, iff (x,y) is a three-valued stable model of $\Phi_{\mathcal{P}}$.
- a *total stable model* of $\mathcal{P}$, iff (x,y) is a two-valued stable model of $\Phi_{\mathcal{P}}$.
- the *well-founded model* of $\mathcal{P}$, iff (x,y) is the well-founded model of $\Phi_{\mathcal{P}}$.

In [17] it is shown that for normal logic programs the partial stable models coincide with the three-valued semantics as defined by [19], the well-founded model coincides with the homonymous semantics as defined by [19,28], and the total stable models coincide with the two-valued (or total) stable models of $\mathcal{P}$.

Example 2. Consider the program $\mathcal{P} = \{q \leftarrow \neg p;\ p \leftarrow \neg q;\ r \leftarrow \neg q;\ r \leftarrow \neg r\}$ (see Example 1). The bilattice of interest is formed by all pairs of subsets of $\{p,q,r\}$.

The partial stable models of $\mathcal{P}$ are $(\emptyset,\{p,q,r\})$, $(\{q\},\{q,r\})$, and $(\{p,r\},\{p,r\})$. In this case, $(\emptyset,\{p,q,r\})$ is well-founded and $(\{p,r\},\{p,r\})$ is total stable.

3. Argumentative Reflections

We now show how non-monotonic formalisms in general, and LP in particular, may be reflected by argumentation frameworks. First, we review some existing results concerning the correspondence between semantical notations in LP and ABA, and then we show how these results may be carried on to further types of LP semantics and other forms of nonmonotonic formalisms, using argumentative reflections of approximated fixpoint concepts.

3.1. Argumentative Characterizations of Logic Programs

The translation of logic programs into assumption-based argumentation has been the subject of several publications (e.g., [5,10,14,23]). The basic idea underlying all of these works is the same: the set of assumptions is made up of negated atoms and the contrary of a negated atom is the positive atom. For such translations it is shown that several argumentation semantics can characterize LP models. For instance, in [5] it is shown that for normal logic programs, complete extensions correspond to partial stable models, the grounded extension corresponds to the well-founded model, preferred extensions correspond to $\leq_i$-maximal partial stable models (also called 'regular'), and stable extensions correspond to two-valued stable models.

The results above are extended in [14] to disjunctive logic programming under stable model semantics. Furthermore, argumentative characterizations of the so-called well-justified [25] and well-founded [29] semantics of general or first-order logic programs with aggregates are provided in [10]. These generalizations are again based on similar representation methods: the assumptions consist of negated atoms and attacks are initiated when the attacking set allows to derive the positive version of the attacked (negated) atom. What changes, however, is the reasoning frame used to determine initiation of attacks. For example, in [14] the reasoning frame is supplemented with rules ensuring the adequate treatment of disjunction. Likewise, in [10], any valid first-order deduction rule is applicable.

Example 3. Consider again the assumption-based framework in Example 1. This is in fact a translation, according to the description of [5] above, of the logic program in Example 2. Indeed, the following semantic elements, indicated in Examples 1 and 2, correspond to the equivalences listed below (For instance, the stable extension of ABF is obtained by the (negation of the) complement of the total stable model of $\mathcal{P}$):

ABF		$\mathcal{P}$	
complete	$\emptyset$, $\{\neg p\}$, $\{\neg q\}$	partial stable	$(\emptyset, \{p,q,r\})$, $(\{q\},\{q,r\})$, $(\{p,r\},\{p,r\})$
grounded	$\emptyset$	well-founded	$(\emptyset, \{p,q,r\})$
stable	$\{\neg q\}$	total stable	$(\{p,r\},\{p,r\})$
peferred	$\{\neg p\}$, $\{\neg q\}$	$\leq_i$-max. stb	$(\{q\},\{q,r\})$, $(\{p,r\},\{p,r\})$

Despite these recent efforts, several questions still remain open. For example, several semantics for disjunctive logic programming have not yet been characterized in assumption-based argumentation. Likewise, three-valued stable models have not been characterized for first-order logic programs with aggregates.

3.2. Argumentative Reflections of Approximated Fixpoint Concepts

We now generalize the ideas discussed previously and show that for any operator over an underlying lattice one may compute its *argumentative reflection*. By this, it will be possible to show a correspondence between the semantical notions from approximation fixpoint theory and those of argumentation-based semantics, provided that the attack relation adequately reflects the operator in question.

The lattice under consideration in what follows is of the form $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$, where $\mathcal{A}$ is some nonempty set.[6] We then denote by $\overline{\mathcal{A}} = \{\overline{A} \mid A \in \mathcal{A}\}$ the set of argumentative reflections of the elements in $\mathcal{A}$. Intuitively and in accordance with the argumentative characterizations described above, $\overline{A}$ can be interpreted as some kind of *absence* of A.[7] Depending on the exact context, this absence can be assumption of falsity, failure to prove, etc. Accordingly, we denote:

- if $\Delta \subseteq \mathcal{A}$, then: $\overline{\Delta} = \{\overline{A} \mid A \in \Delta\}$ and $\sim\Delta = \mathcal{A} \setminus \Delta$.
- if $\Delta \subseteq \overline{\mathcal{A}}$, then: $\underline{\Delta} = \{A \in \mathcal{A} \mid \overline{A} \in \Delta\}$ and $\sim\Delta = \overline{\mathcal{A}} \setminus \Delta$.

We shall say that the lattice $L_{\overline{\mathcal{A}}} = (\wp(\overline{\mathcal{A}}), \subseteq)$ is the *reflection* of $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$. In what follows we shall assume that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ and that $\overline{A} \neq \overline{B}$ for every distinct $A, B \in \mathcal{A}$.

Remark 4. The assumption that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ is meant to assure that the resulting reflecting assumption-based frameworks (Definition 10) will be flat. This assumption holds for the translations of LP discussed above, as well as for the translation of default logic in ABA (see [3]). For normal logic programs, such an assumption is automatically satisfied, since heads of rules contain only positive atoms. When moving to general logic programs, [10] introduces for every atom a new element $\overline{A}$ (originally denoted $\mathsf{not}\,A$) such that $\mathsf{not}\,A \vdash \neg A$. Here we follow a similar idea for a more general case. For the translation of autoepistemic logic from [3], the assumption that $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$ is not warranted. The generalization of the results in this paper for such formalisms is left for future work.

We now turn to the primary concept of representations by argumentation frameworks. The underlying idea is to assume the 'absence' ($\overline{A}$) of any $A \in \mathcal{A}$, unless, on the basis of $C(\mathcal{O})$, some set of assumptions indicates that A holds. Thus, the complete stable operator $C(\mathcal{O})$ should be reflected in the attack relations.

Definition 9. The *argumentative reflection* of an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is given by the framework $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \leadsto \rangle$, where $\Delta \leadsto \overline{A}$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$.

Note that since $\overline{\Delta}$ and $\underline{\Delta}$ are complementary operators on Δ (that is, $\underline{\overline{\Delta}} = \Delta$), moving back and forth between a lattice $L_{\mathcal{A}} = (\wp(\mathcal{A}), \subseteq)$ and an argumentative reflection $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ of an approximation operator $\mathcal{O}$ on $(L_{\mathcal{A}})^2$, is straightforward. In particular, we have:

[6] For instance, $\mathcal{A}$ may be the set of the atomic formulas appearing in a logic program.

[7] For instance, as indicated in the previous section, in LP reflections are the negated atoms of the atoms of the logic program, that is:, $\overline{\mathcal{A}} = \{\neg A \mid A \in \mathcal{A}\}$, where $\neg$ may be a 'negation as failure' connective.

Lemma 1. *Suppose that* $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \rightsquigarrow \rangle$ *reflects* $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$. *Then:*

- *a) for every* $\Delta \subseteq \wp(\overline{\mathcal{A}})$ *it holds that* $\overline{\sim(\underline{\sim\Delta})} = \Delta$ *and for every* $\Delta \subseteq \wp(\mathcal{A})$ *it holds that* $\underline{\sim(\overline{\sim\Delta})} = \Delta$,
- *b) if* (x,y) *is a three-valued stable model of* $\mathcal{O}$, *then* $x = \underline{(\overline{\sim y})^{+}}$.

Proof. We show the first part of (a) (the other one is similar): Since $\underline{\sim\Delta} = \{A \in \mathcal{A} \mid \overline{A} \notin \Delta\}$, we have that $\overline{\sim(\underline{\sim\Delta})} = \{\overline{A} \mid A \notin \{A \in \mathcal{A} \mid \overline{A} \notin \Delta\}\} = \{\overline{A} \mid \overline{A} \in \Delta\} = \Delta$.

For (b), note that since (x,y) is stable, $x = lfp(\mathcal{O}_l(.,\underline{\sim y}) = C(\mathcal{O})(\underline{\sim y})$. Since $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects $\mathcal{O}$, this means that $(\overline{\sim y})^{+} = \{\overline{A} \mid A \in x\}$. Thus, $\underline{(\overline{\sim y})^{+}} = x$. □

Note that Item (b) of the lemma indicates that when we are given an argumentation framework that reflects $\mathcal{O}$, only the y-component determines the stable models of $\mathcal{O}$.

Now we can state the next result, which is the meta-theoretical basis of all the other propositions that follow.

Proposition 1. *Suppose that* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ *reflects an operator* $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$. *Then* Δ *is a complete extension of* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ *iff* $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ *is a consistent three-valued stable model of* $\mathcal{O}$.

Proof. We show that the condition is necessary for being Δ a complete extension of $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$, omitting the other direction due to space restrictions. Suppose that $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ is a three-valued stable model, we show that Δ is complete. First, we note that, by Item (b) of Lemma 1, $\underline{\Delta^{+}} = \underline{(\overline{\sim\Delta})^{+}}$, and so: $(\star)$ $\Delta = \overline{\sim\Delta}$.
(1) Conflict-freeness: Since $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ is consistent, $\underline{\Delta^{+}} \subseteq \underline{\sim\Delta}$, and so $\Delta^{+} \subseteq \sim\Delta$. Thus Δ attacks only elements in its complementary set.
(2) Admissibility: Suppose that there is a set $\Theta \subseteq \overline{\mathcal{A}}$ such that $\Theta \rightsquigarrow \overline{A}$ for some $\overline{A} \in \Delta$. By $(\star)$, $\overline{A} \in \overline{\sim\Delta}$, so in particular, $A \notin \underline{\Delta}$. This means with the stability of $(\underline{\Delta^{+}}, \underline{\sim\Delta})$ that $A \notin lfp(\mathcal{O}_l(.,\underline{\Delta^{+}})$, i.e., $A \notin C(\mathcal{O})(\underline{\Delta^{+}})$. Since $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ reflects $\mathcal{O}$, this means that $\overline{\sim(\underline{\Delta^{+}})} \not\rightsquigarrow \overline{A}$. By definition, $\overline{\sim(\underline{\Delta^{+}})} = \{\overline{A} \mid \overline{A} \notin \Delta^{+}\} = \overline{\mathcal{A}} \setminus \Delta^{+}$. Thus, $\overline{\sim(\underline{\Delta^{+}})} \not\rightsquigarrow \overline{A}$ means that $\overline{\mathcal{A}} \setminus \Delta^{+} \not\rightsquigarrow \overline{A}$, which implies that if for some $\Gamma \subseteq \overline{\mathcal{A}}$, $\Gamma \rightsquigarrow \overline{A}$, then $\Gamma \cap \Delta^{+} \neq \emptyset$. Thus, $\Delta^{+} \cap \Theta \neq \emptyset$, which means that Θ is attacked by Δ, and so Δ defends $\overline{A}$.
(3) Completeness: Similar to the proof of admissibility. □

Proposition 2. *It* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ *reflects an operator* $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$, *then:*

1. (x,y) *is the well-founded model of* $\mathcal{O}$ *iff* $\overline{\sim y}$ *is the grounded extension of* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.
2. (x,y) *is a stable model of* $\mathcal{O}$ *iff* $\overline{\sim y}$ *is a two-valued stable model of* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.
3. (x,y) *is a* $\leq_i$*-maximal three-valued stable model of* $\mathcal{O}$ *iff* $\overline{\sim y}$ *is a preferred extension of* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$.

Proof. We show one direction of the first item. The proofs of the other claims are similar. For the proof we need the following two lemmas:

Lemma 2. *Suppose that* $\mathcal{AF}_{\mathcal{A},\mathcal{O}}$ *reflects an operator* $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$. *Let* (x_1,y_1) *and* (x_2,y_2) *be three-valued stable models of* $\mathcal{O}$. *Then* $(x_1,y_1) \leq_i (x_2,y_2)$ *iff* $\overline{\sim y_1} \cup (\overline{\sim y_1})^{+} \subseteq \overline{\sim y_2} \cup (\overline{\sim y_2})^{+}$.

Lemma 3. *Let $\mathcal{O}$ be an approximating operator over the lattice $(L_{\mathcal{A}})^2$ and let $\mathcal{AF}_{\mathcal{A},\mathcal{O}} = \langle \wp(\overline{\mathcal{A}}), \rightsquigarrow \rangle$ be the argumentative reflection of $\mathcal{O}$. Then $\rightsquigarrow$ is monotonic: If $\Delta \rightsquigarrow \overline{A}$ and $\Delta \subseteq \Theta$, then $\Theta \rightsquigarrow \overline{A}$.*

Suppose now that (x,y) is the well-founded model of $\mathcal{O}$. By Proposition 1 and since the well-founded model is three-valued stable model, $\overline{\sim y}$ is complete. Suppose now that there is some $\Delta \subset \overline{\sim y}$ such that Δ is complete. By Proposition 1, $(\underline{\Delta^+}, \underline{\sim\Delta})$ is a three-valued stable model of $\mathcal{O}$. By Lemma 3, $\Delta \subset \overline{\sim y}$ implies $\Delta^+ \subseteq (\overline{\sim y})^+$. By Lemma 2, this implies that $(\underline{\Delta^+}, \underline{\sim\Delta}) \leq_i (x,y)$. But since $(\underline{\Delta^+}, \underline{\sim\Delta})$ is stable, (x,y) cannot be well-founded. □

4. Applications

In the first part of this section (Section 4.1) we demonstrate the usefulness of the results in Section 3.2 by showing how to obtain an assumption-based framework whose argumentation framework constitutes an argumentative reflection of a given operator. Then, in Section 4.2 we illustrate this in detail using propositional logic programs as defined in Section 2.3.

4.1. Assumption-based Argumentative Reflection

We show how to obtain an ABF whose argumentation framework is an argumentative reflection (Definition 9) of a given operator $\mathcal{O}$. First, we define an appropriate reasoning frame:

Lemma 4. *Let $\mathcal{O}$ be an approximating operator over the lattice $(L_{\mathcal{A}})^2$. Consider the pair $\mathfrak{L}_{\mathcal{O}} = \langle \mathcal{L}, \vdash \rangle$, where $\mathcal{L}$ includes both $\mathcal{A}$ and $\overline{\mathcal{A}}$, and $\vdash$ is defined for $\Delta \subseteq \overline{\mathcal{A}}$ by $\Delta \vdash A$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. Then $\mathfrak{L}_{\mathcal{O}}$ is a reasoning frame, i.e., $\vdash$ is monotonic.*

Definition 10. The *assumption-based argumentative reflection* of an operator $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is given by the assumption-based argumentation framework $\mathsf{ABF}_{\mathcal{A},\mathcal{O}} = \langle \mathfrak{L}_{\mathcal{O}}, \Lambda, - \rangle$, where $\mathfrak{L}_{\mathcal{O}}$ is the reasoning frame defined in Lemma 4, $\Lambda = \overline{\mathcal{A}}$, and the contrariness operator is defined for every $\overline{A} \in \overline{\mathcal{A}}$ by $-\overline{A} = A$.

Proposition 3. *If $\mathcal{O} : (L_{\mathcal{A}})^2 \to (L_{\mathcal{A}})^2$ is approximating (in the sense of Definition 6), then $\mathcal{AF}(\mathsf{ABF}_{\mathcal{A},\mathcal{O}})$ (Definition 4 and 10) reflects $\mathcal{O}$.*

Proof. We have to show that for any $\Delta \cup \{\overline{A}\} \subseteq \overline{\mathcal{A}}$, Δ attacks $\overline{A}$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. By definition, Δ attacks $\overline{A}$ iff $\Delta \vdash -\overline{A}$. Since $-\overline{A} = A$, Δ attacks $\overline{A}$ iff $\Delta \vdash A$. By the definition of $\mathsf{ABF}_{\mathcal{A},\mathcal{O}}$, $\Delta \vdash A$ iff $A \in C(\mathcal{O})(\underline{\sim\Delta})$. □

4.2. Example: Propositional Logic Programming

We now illustrate how the results shown in the previous section can be applied to obtain argumentative characterizations of logic programs as defined in Section 2.3.

Theorem 1. *Let $\mathcal{P}$ be a logic program and let $\overline{\mathsf{Atoms}(\mathcal{P})} = \{\overline{A} \mid A \in \mathsf{Atoms}(\mathcal{P})\}$. Consider the assumption-based framework $\mathsf{ABF}_{\mathcal{P}} = \langle \mathfrak{L}_{\mathcal{P}}, \overline{\mathsf{Atoms}(\mathcal{P})}, - \rangle$, where $-\overline{A} = A$ and $\mathfrak{L}_{\mathcal{P}} = \langle \mathcal{L}_{\mathcal{P}}, \vdash_{\mathcal{P}} \rangle$ is a reasoning frame in which $\mathcal{L}_{\mathcal{P}}$ includes $\overline{\mathsf{Atoms}(\mathcal{P})}$ and the closure under $\neg, \wedge, \vee$ of $\mathsf{Atoms}(\mathcal{P})$, and $\Delta \vdash_{\mathcal{P}} \phi$ iff $\phi \in C(\Psi^l_{\mathcal{P}})(\underline{\sim \Delta})$. For a set $\Delta \subseteq \overline{\mathsf{Atoms}(\mathcal{P})}$ we denote $\Delta^* = (\underline{\Delta^+}, \underline{\sim \Delta})$. Then:*

1. *Δ is a complete extension of $\mathsf{ABF}_{\mathcal{P}}$ iff Δ^* is a partial stable model of $\mathcal{P}$.*
2. *Δ is the grounded extension of $\mathsf{ABF}_{\mathcal{P}}$ iff Δ^* is the well-founded model of $\mathcal{P}$.*
3. *Δ is a stable extension of $\mathsf{ABF}_{\mathcal{P}}$ iff Δ^* is a total stable model of $\mathcal{P}$.*
4. *Δ is a preferred extension of $\mathsf{ABF}_{\mathcal{P}}$ iff Δ^* is a $\leq_i$-maximal partial stable model of $\mathcal{P}$.*

Proof. Since $\mathsf{ABF}_{\mathcal{P}} = \mathsf{ABF}_{\mathsf{Atoms}(\mathcal{P}), \Psi^l_{\mathcal{P}}}$, by Proposition 3, $\mathcal{AF}(\mathsf{ABF}_{\mathcal{P}})$ reflects $\Psi^l_{\mathcal{P}}$. Thus, Item 1 is obtained by Proposition 1 and Definition 8. Items 2, 3, and 4 are obtained in a similar way, using respectively Items 1, 2 and 3 of Proposition 2. □

Remark 5. The results of Theorem 1 were already shown for grounded and stable extensions (i.e., Items 2 and 3) in [10].[8] On the other hand, the correspondence between complete and partial stable extensions (and the analogous correspondence for preferred extensions) was left in [10] as a conjecture. We are now able to confirm this conjecture with Theorem 1.

Remark 6. For normal logic programs $C(\Psi^l_{\mathcal{P}}(\sim \Delta))$ is nothing but the consequence operator $\vdash$ based on Modus Ponens from Example 1. Furthermore, for the general case of propositional logic programs, it can be shown that for consistent sets of assumptions, $C(\Psi^l_{\mathcal{P}}(\sim \Delta))$ is the consequence operator that satisfies Modus Ponens with respect to the rules in the logic program and every valid inference for classical logic (this is the consequence operator used in [10] as described in Section 3.2).

Example 4. Theorem 1 can be illustrated by revisiting Example 1. Indeed, there $\mathsf{ABF}_{\mathcal{P}}$ is the assumption-based argumentative reflection of $\Psi^l_{\mathcal{P}}$ (as in Theorem 1), when restricted to normal logic programs. We see, then, that the semantic equivalences observed in Example 3 are not a coincidence, since they follow from Theorem 1.

Argumentative characterizations similar to those in Theorem 1 can be obtained as corollaries of our results for many other variants of logic programs (such as first-order logic programs, logic programing with aggregates and HEX-programs). Furthermore, the generalization of an argumentative characterizations of semantical alternatives to Reiter's extensions for default logic can also be obtained as a corollary of our main results and the characterization of default logic in approximation fixpoint theory from [8], thus significantly extending the argumentative characterization of Reiter's default extensions in [3].

[8] In fact, in [10] this correspondence is shown for first-order logic programs where the head of a rule can be any propositional formula. Due to space limitations we have restricted ourselves to the propositional case where heads of rules are literals.

5. Conclusion, In View of Related Work

Our results allow for translations of any non-monotonic formalism which has received characterizations in approximation fixpoint theory (and gives rise to flat assumption-based reflections). This includes default logic under the semantics discussed in Section 2.2, as well as many families of logic programming languages under various semantics like those in Section 2.3.

To the best of our knowledge, a general methodology for argumentative characterizations of non-monotonic formalisms has not been suggested before. The connections between approximation fixpoint theory and abstract argumentation where investigated already in [26], where it was shown that abstract dialectical frameworks [4] and Dung's abstract argumentation [9] can be characterized using approximation fixpoint theory. Even though the results of this paper are in a sense complementary to those in [26] (that is, that argumentation theory can capture approximation fixpoint theory), the goal of our paper is somewhat orthogonal: the argumentative characterization of approximation fixpoint theory is to be seen as a mean to obtain a multitude of results on argumentative characterizations of non-monotonic formalisms.

The translation of non-monotonic formalisms into assumption-based argumentation is not only interesting from a theoretical point of view, but also allows for importing methods from one formalism to the other. For example, argumentative characterizations of logic programming have been proven useful for explanation [22,24] and visualization [21] of inferences in logic programming. Our results now open the door for the applications of such techniques to any of the formalisms discussed in this paper. Furthermore, extensions of assumption-based argumentation, such as the integration of priorities [6], can now be combined with the translated formalisms.

In future work, we plan to extend our results to approximation operators that give rise to non-flat ABFs (such as those for autoepistemic logic) and consider non-deterministic operators [18].

References

[1] Christian Antić, Thomas Eiter, and Michael Fink. Hex semantics via approximation fixpoint theory. In *Proceedings of LPNMR'13*, LNCS 8148, pages 102–115. Springer, 2013.

[2] Chitta R Baral and V. S. Subrahmanian. Dualities between alternative semantics for logic programming and nonmonotonic reasoning. *Journal of Automated Reasoning*, 10(3):399–420, 1993.

[3] Andrei Bondarenko, Phan Minh Dung, Robert Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63–101, 1997.

[4] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proceedings of KR'10.* AAAI Press, 2010.

[5] Martin Caminada and Claudia Schulz. On the equivalence between assumption-based argumentation and logic programming. *Artificial Intelligence Research*, 60:779–825, 2017.

[6] Kristijonas Cyras and Francesca Toni. ABA+: assumption-based argumentation with preferences. In *Proceedings of KR'16.* AAAI Press, 2016.

[7] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In *Logic-based*

Artificial Intelligence, volume 597 of *The Springer International Series in Engineering and Computer Science*, pages 127–144. Springer, 2000.

[8] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence*, 143(1):79–122, 2003.

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

[10] Phan Minh Dung, Tran Cao Son, and Phan Minh Thang. Argumentation-based semantics for logic programs with first-order formulae. In *Proceedings of PRIMA'16*, LNCS 9862, pages 43–60. Springer, 2016.

[11] Melvin Fitting. Bilattices are nice things. In *Self Reference*, volume 178 of *CSLI Lecture Notes*, pages 53–77. CLSI Publications, 2006.

[12] Matthew L. Ginsberg. Multi-valued logics: A uniform approach to reasoning in artificial intelligence. *Computer Intelligence*, 4:256–316, 1988.

[13] Jesse Heyninck and Ofer Arieli. On the semantics of simple contrapositive assumption-based argumentation frameworks. In *Proceedings of COMMA'18*, Frontiers in Artificial Intelligence and Applications 305, pages 9–20. IOS Press, 2018.

[14] Jesse Heyninck and Ofer Arieli. An argumentative characterization of disjunctive logic programming. In *Proceedings of EPIA'19*, LNCS 11805, pages 526–538. Springer, 2019.

[15] Victor Marek and Miroslaw Truszczynski. Relating autoepistemic and default logics. In *Proceedings of KR'89*, pages 276–288. Morgan Kaufmann, 1989.

[16] Robert Moore. Semantic considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[17] Nikolay Pelov, Marc Denecker, and Maurice Bruynooghe. Well-founded and stable semantics of logic programs with aggregates. *Theory and Practice of Logic Programming*, 7(3):301–353, 2007.

[18] Nikolay Pelov and Miroslaw Truszczynski. Semantics of disjunctive programs with monotone aggregates – an operator-based approach. In *Proceedings of NMR'04*, pages 327–334, 2004.

[19] Teodor C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.

[20] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[21] Claudia Schulz. Graphical representation of assumption-based argumentation. In *Proceedings of AAAI'15*, pages 4204–4205. AAAI Press, 2015.

[22] Claudia Schulz, Ken Satoh, and Francesca Toni. Characterising and explaining inconsistency in logic programs. In *Proceedings of LPNMR'15*, LNCS 9345, pages 467–479. Springer, 2015.

[23] Claudia Schulz and Francesca Toni. Logic programming in assumption-based argumentation revisited-semantics and graphical representation. In *Proceedings of AAAI'15*, pages 1569–1575. AAAI Press, 2015.

[24] Claudia Schulz and Francesca Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59âĂŞ110, 2016.

[25] Yi-Dong Shen, Kewen Wang, Thomas Eiter, Michael Fink, Christoph Redl, Thomas Krennwallner, and Jun Deng. FLP answer set semantics without circular justifications for general logic programs. *Artificial Intelligence*, 213:1–41, 2014.

[26] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[27] Hannes Strass and Johannes Peter Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, 226:34–74, 2015.

[28] Allen Van Gelder, Kenneth A Ross, and John S Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, 1991.

[29] Yisong Wang, Fangzhen Lin, Mingyi Zhang, and Jia-Huai You. A well-founded semantics for basic logic programs with arbitrary abstract constraint atoms. In *Proceedings of AAAI'12*, pages 835–841. AAAI Press, 2012.