# Flexible Approximators for Approximating Fixpoint Theory

Fangfang Liu[1], Yi Bi[2], Md. Solimul Chowdhury[3], Jia-Huai You[3], Zhiyong Feng[2]

[1] School of Computer Engineering and Science, Shanghai University, China
[2] School of Computer Science and Technology, Tianjin University, China
[3] Department of Computing Science, University of Alberta, Canada

**Abstract.** Approximation fixpoint theory (AFT) is an algebraic framework for the study of fixpoints of operators on bilattices, which has been applied to the study of the semantics for a number of nonmonotonic formalisms. A central notion of AFT is that of *stable revision* based on an underlying approximating operator (called approximator), where the negative information used in fixpoint computation is by default. This raises a problem in systems that combine different formalisms, where both default negation and established negation may be present in reasoning. In this paper we extend AFT to allow more flexible approximators. The main idea is to formulate and propose ternary approximators, of which traditional binary approximators are a special case. The extra parameter allows separation of two kinds of negative information, by entailment and by default, respectively. The new approach is motivated by the need to integrate different knowledge representation and reasoning (KRR) systems, in particular to support combined reasoning by nonmonotonic rules with ontologies. However, this small change by allowing flexible approximators raises a mathematical question - whether the resulting AFT is a sound fixpoint theory. The main result of this paper is a proof that answers this question positively.

## 1 Introduction

AFT, also known as the theory of consistent approximations, is a powerful framework for the study of fixpoints for nonmonotonic logics, which has applied to study semantics for various types of logic programs [1–4, 6, 14], argumentation systems [7, 17], and default and autoepistemic logic [5]. Under this theory, the semantics of a logic program is defined by respective fixpoints closely related to an *approximator* on a bilattice. The approach is highly general as it only depends on mild conditions on approximators. The *well-founded fixpoint* of an approximator defines a well-founded semantics (WFS) and *exact stable fixpoints* define an answer set (or stable) semantics. As different approximators may represent different intuitions, AFT provides a way to treat semantics uniformly and allows to explore alternatives by different approximators. The properties of a semantics hold even without a concrete approximator. For example, the least fixpoint approximates all other fixpoints and, mathematically this property holds for any approximator. Practically, since the WFS is generally easier to compute, it can be employed as an approximation or as a mechanism of constraint propagation in computing exact stable fixpoints.

In this paper we address a theoretical question that arises in appyling the current AFT to integrating different KRR formalisms, in particular to the so-called FOL-programs, which are combined knowledge bases $KB = (L, \Pi)$, where $L$ is a theory of a decidable fragment of first-order logic and $\Pi$ a set of rules possibly containing some arbitrary first-order formulas. This is one of the recent approaches to combining answer set programming (ASP) with description logics (DLs) [8, 12, 13, 15, 20, 21]. As an illustration, assume $L$ contains a formula that says all students are entitled to educational discount, $\forall x\ St(x) \supset EdDiscount(x)$. Suppose in an application anyone who is not employed but registered for a class is given the benefit of a student. We can write a rule: $St(X) \leftarrow TakeClass(X, Y), not\ HasJob(X)$. Thus, that such a person enjoys educational discount can be inferred directly from knowledge base $L$. Note that unemployment is typically verified by default, e.g., by lack of tax record for income.

In AFT, to define a semantics for a nonmonotonic logic is to define an approximator, which is required to be monotone on the underlying bilattice. Then the well-known Knaster-Tarski fixpoint theory [18] can be generalized to chain-complete posets. The least fixpoint of an approximator $\mathcal{A}$ is called the *Kripke-Kleene fixpoint of* $\mathcal{A}$. In logic programming, this fixpoint corresponds to what is typically called Kripke-Kleene semantics due to the fact that it does not compute *unfounded atoms*. E.g., given a logic program $P = \{a \leftarrow a\}$, its Kripke-Kleene semantics is that $a$ is *undefined* or *unknown* in 3-valued logic, while the more desirable well-founded and stable semantics both assign $a$ to *false*, since making it false does not invalidate the rule.

To be able to capture unfounded atoms is a key feature of ASP semantics. In the context of ASP, a pair $(u, v)$ on the bilattice built from the power set of atoms under the subset relation is viewed as a 3-valued interpretation (also called a *partial interpretation*): atoms in $u$ are true, those in $v$ are *possibly true*, thus those not in $v$ are false and the atoms that are not true but possibly true take the truth value *undefined* (or called *unknown*). Roughly speaking, to capture unfounded atoms is to compute those atoms that can be assigned to false without invalidating any rule. In AFT, this is accomplished by what is called a *stable revision operator*, which maps a given pair $(u, v)$ to a new pair by two least fixpoints, denoted $(lfp(\mathcal{A}^1(\cdot, v)), lfp(\mathcal{A}^2(u, \cdot)))$, where $\mathcal{A}^1(\cdot, v)$ and $\mathcal{A}^2(u, \cdot)$ are the projection operators of $\mathcal{A}$ on its first and second components, respectively. Intuitively, given $(u, v)$, $lfp(\mathcal{A}^1(\cdot, v))$ computes the atoms that must be true and $lfp(\mathcal{A}^2(u, \cdot))$ computes those that are possibly true. Then, any atoms that are not possibly true are false.

However, a drawback in the construction of $lfp(\mathcal{A}^2(u, \cdot))$ is that the operator $\mathcal{A}^2(u, \cdot)$ has no access to $v$. As a result, in applying AFT to logic programs, two kinds of false atoms are mixed together and used in the computation of $lfp(\mathcal{A}^2(u, \cdot))$. E.g., to compute $\mathcal{A}^2(u, u)$, the first step in the iterative construction of $lfp(\mathcal{A}^2(u, \cdot))$[1], two kinds of false atoms in $(u, u)$ are used in the computation: for an atom $a$, if $a \notin v$ then it is false by the given partial interpretation $(u, v)$; if $a \notin u$ and $a \in v$ ($a$ is not true but possibly true) then $a$ being false in $(u, u)$ (since $a \notin u$) is made by default.

[1] By the Knaster-Tarski fixpoint theory, the least fixpoint can be computed iteratively from the least element of the underlying lattice - in this case, $u$ is the least element in the lattice domain represented by the interval $[u, \top]$.

Though this notion of stable revision works well for many nonmonotonic logics, problem arises in combining ASP with classic logic, e.g., for FOL-programs, where one faces two undesirable possibilities in computing $\mathcal{A}^2(u, v)$ - *under estimate* or *over estimate* of negative knowledge.

To illustrate the point of "over estimate", assume an FOL-program $KB = (L, \Pi)$ where $L = \{P(a)\}$ and $\Pi = \{P(a) \leftarrow P(a);\ P(b) \leftarrow F\}$, where $F$ is some arbitrary formula. Stable revision starts from the least element on the underlying bilattice, which is $(\bot, \top)$ (in this context, $\bot$ denotes the empty set and $\top$ the set of all atoms). Given $(\bot, \top)$, we need to compute $(lfp(\mathcal{A}^1(\cdot, \top)), lfp(\mathcal{A}^2(\bot, \cdot)))$, where the first step in computing the latter is to compute $\mathcal{A}^2(\bot, \bot)$.[2] If approximator $\mathcal{A}$ is defined in a way that $\mathcal{A}^2(\bot, \bot)$ uses $L$ and partial interpretation $(\bot, \bot)$ as premises, we will have inconsistent premises, as $P(a)$ is false in $(\bot, \bot)$ which contradicts $L$. If we allow absurdity to lead to the derivation of $P(b)$, we may then compute an unintended conclusion: $P(b)$ is possibly true. We will have more to say about under estimate later, in Section 3.

This paper addresses the above issue by proposing the notion of *ternary* approximators, where the extra parameter holds the information on (already computed) negative atoms. Though this increases representation flexibility, a theoretical question is whether the resulting AFT is a sound fixpoint theory, in that a monotone approximator is guaranteed to possess fixpoints and a least fixpoint. The main result of this paper is a mathematical development that shows the resulting AFT is indeed sound.

The next section introduces background on bilattices, followed by Section 3 on extending AFT. As an example, Section 4 applies the extended AFT to FOL-programs. This is followed by a discussion of related work and future directions.

## 2 Background

We assume familiarity with Knaster-Tarski fixpoint theory [18]. Briefly, a *lattice* $\langle \mathcal{L}, \leq \rangle$ is a *poset* in which every two elements have a least upper bound (lub) and a greatest lower bound (glb). A *chain* in a poset is a linearly ordered subset of $\mathcal{L}$. A poset $\langle \mathcal{L}, \leq \rangle$ is *chain-complete* if it contains a least element $\bot$ and every chain $C \subseteq \mathcal{L}$ has a lub in $\mathcal{L}$. A lattice $\langle \mathcal{L}, \leq \rangle$ is *complete* if every subset $S \subseteq \mathcal{L}$ has a lub and a glb.

Let $\langle \mathcal{L}, \leq \rangle$ be a complete lattice. The structure $\langle \mathcal{L}^2, \leq, \leq_p \rangle$ denotes the induced (product) bilattice, where $\leq_p$ is called the *precision order* and defined as: for all $x, y$, $x', y' \in \mathcal{L}$, $(x, y) \leq_p (x', y')$ if $x \leq x'$ and $y' \leq y$. The $\leq_p$ ordering is a complete lattice ordering on $\mathcal{L}^2$. In this paper, we refer to a lattice $\langle \mathcal{L}, \leq \rangle$ simply by $\mathcal{L}$, which is always assumed to be complete, and denote the induced bilattice by $\mathcal{L}^2$.

We say that a pair $(x, y) \in \mathcal{L}^2$ is *consistent* if $x \leq y$, *inconsistent* otherwise, and *exact* if $x = y$. We denote the set of all consistent pairs by $\mathcal{L}^c$. Note that the restriction to $\mathcal{L}^c$ does not form a sublattice. It is a chain-complete poset, with maximal elements being exact pairs. A consistent pair $(x, y) \in \mathcal{L}^c$ defines an *interval*, denoted by $[x, y]$, which defines the set $\{z \mid x \leq z \leq y\}$. A consistent pair $(x, y)$ in $\mathcal{L}^c$ can be seen as an *approximation* of every $z \in \mathcal{L}$ such that $z \in [x, y]$. In this sense, the precision order $\leq_p$ corresponds to the precision of approximation, while an exact pair approximates the only element in it.

[2] again, because $\bot$ is the least element of the domain $[\bot, \top]$.

An operator $\mathcal{O}$ on a complete lattice or a chain-complete poset $\mathcal{L}$ is monotone if for all $x, y \in \mathcal{L}$, $x \leq y$ implies $\mathcal{O}(x) \leq \mathcal{O}(y)$. Such a monotone operator possesses fixpoints and a least fixpoint. We denote the least fixpoint of $\mathcal{O}$ by *lfp*$(\mathcal{O})$. An element $x \in \mathcal{L}$ is a pre-fixpoint of $\mathcal{O}$ if $\mathcal{O}(x) \leq x$; it is a post-fixpoint of $\mathcal{O}$ if $x \leq \mathcal{O}(x)$.

## 3 An Extended Theory of Approximation

The original AFT is built on $\mathcal{L}^c$. We generalize AFT to $\mathcal{L}^2$ by addressing two issues. The first is on the notion of approximator, and the second is on enriching algebraic manipulation by a stable revision operator.

An *approximator* $\mathcal{A}$ is a $\leq_p$-monotone operator on $\mathcal{L}^2$ that approximates an operator $\mathcal{O}$ on $\mathcal{L}$. In the original theory, it is required that $\mathcal{A}(z, z) = (\mathcal{O}(z), \mathcal{O}(z))$, for all $z \in \mathcal{L}$; i.e., $\mathcal{A}$ extends $\mathcal{O}$ on all exact pairs. This is desired if we only deal with consistent pairs. However, if $\mathcal{A}(z, z)$ is inconsistent, it is possible that $\mathcal{O}(z)$ lies outside of $[x, y]$, as an exact pair in general may not be a maximal element in $\mathcal{L}^2$. This leads to the first attempt to define an approximator on $\mathcal{L}^2$ in Def. 1 below as presented in [2].

We shall remark that the authors of [2] formulated an extension that treats all pairs in $\mathcal{L}$. Since a primary application area of our work is on integrating different formalisms where inconsistency frequently arises, it is natural to build our work on top of [2].

**Definition 1.** *Let $\mathcal{O}$ be an operator on $\mathcal{L}$. We say that $\mathcal{A} : \mathcal{L}^2 \to \mathcal{L}^2$ is an* approximator *of $\mathcal{O}$ iff the following conditions are satisfied:*

- *For all $x \in \mathcal{L}$, if $\mathcal{A}(x, x)$ is consistent then $\mathcal{A}(x, x) = (\mathcal{O}(x), \mathcal{O}(x))$.*
- *$\mathcal{A}$ is $\leq_p$-monotone.*

*Example 1.* (Borrowed from [2]) To see why the consistency condition "$\mathcal{A}(x, x)$ is consistent" in the definition is critical, consider a complete lattice where $\mathcal{L} = \{\bot, \top\}$ and $\leq$ is defined as usual. Let $\mathcal{O}$ be the identify function on $\mathcal{L}$. Then we have two fixpoints, $\mathcal{O}(\bot) = \bot$ and $\mathcal{O}(\top) = \top$. Let $\mathcal{A}$ be an identity function on $\mathcal{L}^2$ everywhere except $\mathcal{A}(\top, \top) = (\top, \bot)$. Thus, $\mathcal{A}(\top, \top)$ is inconsistent. It is easy to check that $\mathcal{A}$ is $\leq_p$-monotone. Since $\mathcal{A}(\bot, \bot) = (\mathcal{O}(\bot), \mathcal{O}(\bot))$, and $(\bot, \bot)$ is the only exact pair such that $\mathcal{A}(\bot, \bot)$ is consistent, $\mathcal{A}$ is an approximator of $\mathcal{O}$, according to the above definition. But $\mathcal{A}(\top, \top) \neq (\mathcal{O}(\top), \mathcal{O}(\top))$, even though $\mathcal{O}(\top) = \top$. If the consistency condition is not imposed on the definition, mappings like the operator $\mathcal{A}$ above would be ruled out as approximators, which means we fail to accommodate inconsistencies as we set out to do.[3]

A central idea of AFT is the notion of *stable revision operator*, denote by $St_{\mathcal{A}}$, for an approximator $\mathcal{A}$. The goal is to determine *persistently reachable* elements as well as *non-reachable* ones in a chain-complete poset, so that a $\leq_p$-monotone operator $St_{\mathcal{A}}$ has fixpoints and a least fixpoint. The latter is called the *well-founded fixpoint* of $\mathcal{A}$ and

[3] This example specifies a system in which states are represented by a pair of factors - high and low. Here, all states are *stable* except the one in which both factors are high. This state may be transmitted to an "inconsistent state" with the first factor high and the second low. This state is the only inconsistent one, and it itself is stable.

the fixpoints of $St_{\mathcal{A}}$ are called the *stable fixpoints* of $\mathcal{A}$. Let us denote by $\mathcal{A}^1$ and $\mathcal{A}^2$ the projection of an operator $\mathcal{A}$ on $\mathcal{L}^2$ on its first and second components, respectively, i.e., $\mathcal{A}^1(\cdot, v)$ is $\mathcal{A}$ with $v$ fixed, and $\mathcal{A}^2(u, \cdot)$ is $\mathcal{A}$ with $u$ fixed.

A consistent pair $(u, v)$ can be viewed as an approximation to any exact pair in the interval $[u, v]$, where $u$ is a *lower estimate* and $v$ an *upper estimate*. The operator $St_{\mathcal{A}}(u, v)$ aims at generating a new pair of lower and upper estimates by the respective fixpoint constructions, as expressed by

$$St_{\mathcal{A}}(u, v) = (\mathit{lfp}(\mathcal{A}^1(\cdot, v)), \mathit{lfp}(\mathcal{A}^2(u, \cdot)). \tag{1}$$

Since approximator $\mathcal{A}$ is $\leq_p$-monotone, so is operator $St_{\mathcal{A}}$, whose least fixpoint can be computed from the least element $(\bot, \top)$ in the given bilattice.

Now let us extend the approach to $\mathcal{L}^2$. It is clear that the operator $\mathcal{A}^1(\cdot, v)$ is defined on $\mathcal{L}$, and if $\mathcal{A}$ is $\leq_p$-monotone on $\mathcal{L}^2$ then $\mathcal{A}^1(\cdot, v)$ is monotone (i.e., $\leq$-monotone) on $\mathcal{L}$. As $\mathcal{L}$ is a complete lattice, according to the Knaster-Tarski fixpoint theorem, $\mathit{lfp}(\mathcal{A}^1(\cdot, v))$ is well-defined. However, there are two problems with $\mathit{lfp}(\mathcal{A}^2(u, \cdot))$. When $St_{\mathcal{A}}(u, v)$ is applied:

1. The operator $\mathcal{A}^2(u, \cdot)$ has no access to $v$, which restricts how an approximator may be defined; and
2. It is possible that $\mathcal{A}^2(u, \cdot) \notin [u, \top]$ and if so, $\mathcal{A}^2(u, \cdot)$ is not an operator on $[u, \top]$ and $\mathit{lfp}(\mathcal{A}^2(u, \cdot))$ is ill-defined.

As remarked in Introduction, without access to $v$, we may either *under estimate*, or *over estimate*, negative knowledge. There, we illustrated the problem with *over estimate*. Here, let us illustrate the problem with *under estimate*. In the next example, we sketch the inferences only intuitively (as we are not in a position to provide all relevant definitions); we will come back to this example in Section 4 (cf. Example 3) once the underlying approximator is defined.

*Example 2.* Consider an FOL-program $KB = (L, \Pi)$ where

$L = \{\forall x\, C(x) \supset (A(x) \lor D(x))\}$ and $\Pi$ has the following rules:

$A(a) \leftarrow A(a).$ $\quad B(a) \leftarrow \mathit{not}\ A(a).$ $\quad D(a) \leftarrow \mathit{not}\ B(a).$
$C(a) \leftarrow \mathit{not}\ C'(a).$ $\quad C'(a) \leftarrow \mathit{not}\ C(a).$

From $L$, if $A(a)$ and $D(a)$ are false, then $C(a)$ must be false thus it should not be possibly true. Thus, a condition on deriving a possibly true atom is that its negation is not entailed by $L$ along with already computed (in this case, negative) information, which is computed as follows: $A(a)$ is false by closed world reasoning. Then, we derive $B(a)$ which leads to the inference that $D(a)$ is false. In terms of stable revision, we have the following sequence starting with the least element $(\bot, \top)$ (for brevity, we write $a$ for $A(a)$, $b$ for $B(a)$, and so on):

$$(\bot, \top) \Rightarrow (\bot, \{c, c', b, d\}) \Rightarrow (\{b\}, \{c, c', b, d\}) \Rightarrow (\{b\}, \{c, c', b\}) \Rightarrow (\{b\}, \{c', b\}) \Rightarrow \ldots$$

where, e.g., atoms $C(a), C'(a), B(a), D(a)$ in the second pair are possibly true, due to possible derivations by rules. In the last pair, $C(a)$ is not possibly true because $\neg C(a)$

is entailed by $L$ and the preceding partial interpretation. Without access to $v$ of $(u, v)$ in computing *lfp*$(\mathcal{A}^2(u,\cdot))$ of (1), we would not be able to infer $\neg C(a)$ hence block the derivation of $C(a)$ for being possibly true. This would lead to a problematic situation where on the one hand $C(a)$ is possibly true and on the other it is provably false.

To tackle this problem, we generalize stable revision by adding an extra parameter $v$ in the definition of $\mathcal{A}$, i.e., we now define $\mathcal{A}$ as a ternary operator $\mathcal{A} : \mathcal{L}^3 \rightarrow \mathcal{L}^2$, which is called by $St_{\mathcal{A}}(u, v)$ with the first parameter fixed to $v$; i.e., it is the operator $\mathcal{A}(v, \cdot, \cdot)$. Let us alternatively write it as $\mathcal{A}_v$. Then, the expression in (1) becomes

$$St_{\mathcal{A}}(u, v) = (\mathit{lfp}(\mathcal{A}_v^1(\cdot, v)), \mathit{lfp}(\mathcal{A}_v^2(u, \cdot)) \tag{2}$$

where the least fixpoints are constructed by respective sequences

$$x_0 = \bot, x_1 = \mathcal{A}_v^1(x_0, v), ..., x_{\alpha+1} = \mathcal{A}_v^1(x_\alpha, v), ... \tag{3}$$
$$y_0 = u, y_1 = \mathcal{A}_v^2(u, y_0), ..., y_{\alpha+1} = \mathcal{A}_v^2(u, y_\alpha), .... \tag{4}$$

Clearly, the subscript $v$ in $\mathcal{A}_v^1(x_i, v)$ does not add any new information, as $v$ is already a parameter of the operator,[4] but $v$ in $\mathcal{A}_v^2(u, y_i)$ $(i \geq 0)$ does.

We now formalize this new notion of approximator.

**Definition 2.** *Let $\mathcal{O}$ be an operator on $\mathcal{L}$. We say that $\mathcal{A} : \mathcal{L}^3 \rightarrow \mathcal{L}^2$ is an approximator of $\mathcal{O}$ iff the following conditions are satisfied:*

*(i) For all $v \in \mathcal{L}$, if $\mathcal{A}_v(v, v)$ is consistent, then $\mathcal{A}_v(v, v) = (\mathcal{O}(v), \mathcal{O}(v))$.*
*(ii) For all $v \in \mathcal{L}$, $\mathcal{A}_v$ is $\leq_p$-monotone.*
*(iii) For all $v, v' \in \mathcal{L}$ such that $v \leq v'$, and all $(x, y) \in \mathcal{L}^2$, $\mathcal{A}_{v'}(x, y) \leq_p \mathcal{A}_v(x, y)$.*

Conditions (i) and (ii) are similar to those in Def. 1, and (iii) ensures no loss of approximation accuracy with smaller $v$ (which is more informed in what are not in $v$).[5] It leads to the notion of monotonicity over different, fixed values $v$ in $\mathcal{A}_v$.

**Lemma 1.** *Let $\mathcal{A} : \mathcal{L}^3 \rightarrow \mathcal{L}^2$ be an approximator, and $v, v' \in \mathcal{L}$ s.t. $v \leq v'$. For all $(x, y), (x', y') \in \mathcal{L}^2$, if $(x, y) \leq_p (x', y')$, then $\mathcal{A}_{v'}(x, y) \leq_p \mathcal{A}_v(x', y')$.*

*Proof.* We can show $\mathcal{A}_{v'}(x, y) \leq_p \mathcal{A}_v(x, y) \leq_p \mathcal{A}_v(x', y')$. The first inequality is by part (iii) of Def. 2 and the second by $\leq_p$–monotonicity of $\mathcal{A}_v$. □

We will build our proposal above to an existing solution to the second problem provided in [2]. Essentially, we need to replace the notion of approximators in [2], by that of ternary approximators formulated above. Mathematically, this is not a trivial process. We now give a detailed mathematical development.

In [6], a desirable property, called $\mathcal{A}$-*reliability*, is introduced to ensure that $\mathcal{A}^2(u, \cdot)$ is an operator on $[u, \top]$. Let us generalize this to $\mathcal{L}^2$. Given an approximator $\mathcal{A}$, $(u, v) \in \mathcal{L}^2$ is called $\mathcal{A}$-*reliable* if $(u, v) \leq_p \mathcal{A}_v(u, v)$.

[4] Note that we never need a parameter to carry (already computed) true atoms, as in computing *lfp*$(\mathcal{A}^1(\cdot, v))$ we do not make default assumptions, and the monotonicity of the operator $\mathcal{A}^1$ guarantees that any previously computed true atoms are derived again.

[5] For example, if $v$ is a set of possibly true atoms, smaller $v$ means more atoms that are false.

However, this property is not strong enough. For example, consider Example 1 again: let $\mathcal{L} = \{\bot, \top\}$ and $\mathcal{A}_x$, for any $x \in \mathcal{L}$, be an identity mapping everywhere except that $\mathcal{A}_\top(\top, \top) = (\top, \bot)$. It can be seen that all $(u, v) \in \mathcal{L}^2$ are $\mathcal{A}$-reliable, but $\mathcal{A}^2_\top(u, \cdot)$ is not defined on $[u, \top]$; e.g., when $(u, v)$ is $(\top, \top)$, we have $\mathcal{A}^2_\top(\top, \top) = \bot$, which is outside the interval $[\top, \top]$.

To ensure that $\mathcal{A}^2_v(u, \cdot)$ is defined on $[u, \top]$, it is sufficient that $A^2_v(u, u) \geq u$ holds, i.e., the first application of the operator $\mathcal{A}^2_v(u, \cdot)$ in (4) yields an element in $[u, \top]$.

**Lemma 2.** *Let $\mathcal{A} : \mathcal{L}^3 \to \mathcal{L}^2$ be an approximating operator. For any $(u, v) \in \mathcal{L}^2$, if $A^2_v(u, u) \geq u$, then for every $z \in [u, \top]$, $\mathcal{A}^2_v(u, z) \in [u, \top]$.*

*Proof.* We have $u \leq A^2_v(u, u) \leq A^2_v(u, z)$. The first inequality is by the condition in the lemma and the second by $(u, z) \leq_p (u, u)$ and $\leq_p$-monotonicity of $\mathcal{A}_v$. □

Question remains as what if the condition $A^2_v(u, u) \geq u$ does not hold, in which case $\mathcal{A}^2_v(u, \cdot)$ is not an operator on $[u, \top]$. To resolve this issue, let us consider the operator $\mathcal{A}^2_v(u, \cdot)$ on $[\bot, \top]$. $\mathcal{A}^2_v(u, \cdot)$ is clearly defined on $[\bot, \top]$. Furthermore, since $\mathcal{A}_v$ is $\leq_p$-monotone on $\mathcal{L}^2$, $\mathcal{A}^2_v(u, \cdot)$ is $\leq$-monotone on $[\bot, \top]$. To verify, $\forall y \in [\bot, \top]$ and $\forall y,' y' \in [\bot, \top]$ such that $y' \leq y$, from $\leq_p$-monotonicity of $\mathcal{A}_v$, we have $\mathcal{A}_v(x, y) \leq_p \mathcal{A}_v(x, y')$, it follows $\mathcal{A}_v(u, y) \leq_p \mathcal{A}_v(u, y')$ and thus $\mathcal{A}^2_v(u, y') \leq \mathcal{A}^2_v(u, y)$.

We are now in a position to define the notion of stable revision.

**Definition 3.** *Let $\mathcal{A} : \mathcal{L}^3 \to \mathcal{L}^2$ be an approximator of some operator on $\mathcal{L}$.* Stable revision operator $St_\mathcal{A} : \mathcal{L}^2 \to \mathcal{L}^2$ *is defined as:*

$$St_\mathcal{A}(u, v) = \begin{cases} (\mathit{lfp}(\mathcal{A}^1_v(\cdot, v)), \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))) \text{ where } \mathcal{A}^2_v(u, \cdot) \text{ is on } [u, \top] & u \leq A^2_v(u, u) \\ (\mathit{lfp}(\mathcal{A}^1_v(\cdot, v)), \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))) \text{ where } \mathcal{A}^2_v(u, \cdot) \text{ is on } [\bot, \top] & \text{otherwise} \end{cases}$$

Before we show the main results of this section, let us introduce and extend another desirable property given in [6]. An element $(u, v) \in \mathcal{L}^2$ is called $\mathcal{A}$*-prudent* if $u \leq \mathit{lfp}(\mathcal{A}^1_v(\cdot, v))$. The reason for this property is that in general there is no guarantee that $\mathit{lfp}(\mathcal{A}^1_v(\cdot, v))$ improves $u$, which is now ensured under $\mathcal{A}$-prudence.

Let us denote by $\mathcal{L}^{rp}$ the set of all $\mathcal{A}$-reliable and $\mathcal{A}$-prudent pairs in $\mathcal{L}^2$.

The first lemma below establishes the chain property of $St_\mathcal{A}$ on $\mathcal{L}^{rp}$, while the next shows its $\leq_p$-monotonicity. For brevity, we may write $St$ for $St_\mathcal{A}$.

**Lemma 3.** *For any pair $(u, v) \in \mathcal{L}^{rp}$, let St(u,v) = $(u', v')$. Then, we have that $(u, v) \leq_p (u', v')$, and $(u', v')$ is $\mathcal{A}$-reliable and $\mathcal{A}$-prudent.*

*Proof.* (Sketch) By $\mathcal{A}$-prudence, we have $u \leq \mathit{lfp}(\mathcal{A}^1_v(\cdot, v))$. To show $v' \leq v$, by $\mathcal{A}_v$-reliability, we have $\mathcal{A}^2_v(u, v) \leq v$, i.e, $v$ is a pre-fixpoint of $\mathcal{A}^2_v(u, \cdot)$. It follows that if $\mathcal{A}^2_v(u, \cdot)$ is an operator on $[u, \top]$ then it must be the case that $v \in [u, \top]$, thus $\mathit{lfp}(\mathcal{A}^2_v(u, \cdot)) \leq v$. If $\mathcal{A}^2_v(u, \cdot)$ is an operator on $[\bot, \top]$, surely $v \in [\bot, \top]$, then we still have $\mathit{lfp}(\mathcal{A}^2_v(u, \cdot)) \leq v$. It then follows $(u, v) \leq_p (u', v')$.

For the second assertion, as $u' = \mathit{lfp}(A^1_v(\cdot, v)) = A^1_v(u', v)$, we have $A^1_v(u', v) \leq A^1_v(u', v')$ by $v' \leq v$. Similarly, as $v' = \mathit{lfp}(\mathcal{A}^2_v(u, \cdot)) = A^2_v(u, v')$, where $\mathcal{A}^2_v(u, \cdot)$ is either an operator on $[u, \top]$ or an operator on $[\bot, \top]$, we have $\mathcal{A}^2_v(u', v') \leq \mathcal{A}^2_v(u, v')$

by $u \leq u'$, thus $(u', v') \leq_p \mathcal{A}_v(u', v')$. Next, by condition (iii) of Def. 2, we have $(u', v') \leq_p \mathcal{A}_v(u', v') \leq \mathcal{A}_{v'}(u', v')$.

Finally, for $\mathcal{A}$-prudence, that $u' \leq \mathit{lfp}(\mathcal{A}^1_{v'}(\cdot, v'))$ can be proved by transfinite induction on the sequences $\mathit{lfp}(\mathcal{A}^1_v(\cdot, v))$ and $\mathit{lfp}(\mathcal{A}^1_{v'}(\cdot, v'))$. □

**Lemma 4.** *For any pairs $(u, v), (u', v') \in \mathcal{L}^{rp}$ such that $(u, v) \leq_p (u', v')$, we have $St(u, v) \leq_p St(u', v')$.*

*Proof.* (Sketch) In all cases, we can show $\mathit{lfp}(\mathcal{A}^1_v(\cdot, v)) \leq_p \mathit{lfp}(\mathcal{A}^1_{v'}(\cdot, v'))$ by transfinite induction on the two sequences. To show $\mathit{lfp}(\mathcal{A}^2_{v'}(u', \cdot)) \leq \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))$, we need to consider four cases: (i) $\mathcal{A}^2_v(u, \cdot)$ and $\mathcal{A}^2_{v'}(u', \cdot)$ are defined on $[u, \top]$ and $[u', \top]$ respectively, (ii) both $\mathcal{A}^2_v(u, \cdot)$ and $\mathcal{A}^2_{v'}(u', \cdot)$ are defined on $[\bot, \top]$ (not in the subdomains as in (i)), (iii) $\mathcal{A}^2_v(u, \cdot)$ is defined on $[u, \top]$ while $\mathcal{A}^2_{v'}(u', \cdot)$ is defined on $[\bot, \top]$, and (iv) $\mathcal{A}^2_v(u, \cdot)$ defined on $[\bot, \top]$ while $\mathcal{A}^2_{v'}(u', \cdot)$ defined on $[u', \top]$.

Let $v_f = \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))$ and $v'_f = \mathit{lfp}(\mathcal{A}^2_{v'}(u', \cdot))$. We then have $v_f = \mathcal{A}^2_v(u, v_f) \geq \mathcal{A}^2_v(u', v_f) \geq \mathcal{A}^2_{v'}(u', v_f)$ by $u \leq u'$ and $v' \leq v$; i.e., $v_f$ is a pre-fixpoint of $\mathcal{A}^2_{v'}(u', \cdot)$. On the other hand, we have $v'_f = \mathcal{A}^2_{v'}(u', v'_f) \leq \mathcal{A}^2_v(u, v'_f)$, i.e., $v'_f$ is a post-fixpoint of $\mathcal{A}^2_v(u, \cdot)$. For case (i), as $v'_f$ is in the domain of $\mathcal{A}^2_{v'}(u', \cdot)$, it is also in the domain of $\mathcal{A}^2_v(u, \cdot)$ by $u \leq u'$; it follows $v'_f \leq \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))$ as it is a post-fixpoint, thus $\mathit{lfp}(\mathcal{A}^2_{v'}(u', \cdot)) \leq \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))$. For case (ii), as $\mathcal{A}^2_v(u, \cdot)$ and $\mathcal{A}^2_{v'}(u', \cdot)$ are both defined on $[\bot, \top]$, that $v'_f \leq v_f$ naturally follows. For case (iii), since $v_f$ is surely in the domain of $\mathcal{A}^2_{v'}(u', \cdot)$ and $v_f$ is a pre-fixpoint of $\mathcal{A}^2_{v'}(u', \cdot)$, it follows $\mathit{lfp}(\mathcal{A}^2_{v'}(u', \cdot)) \leq v_f$. For case (iv), also as $v'_f$ is in the domain of $\mathcal{A}^2_v(u, \cdot)$ and $v'_f$ a post-fixpoint of $\mathcal{A}^2_v(u, \cdot)$, we have $v'_f \leq \mathit{lfp}(\mathcal{A}^2_v(u, \cdot))$. We therefore conclude $St(u, v) \leq_p St(u', v')$. □

Let $C$ be a chain in $\mathcal{L}^{rp}$. Denote by $C^1$ and $C^2$, respectively, the projection of $C$ on its first and second elements. It is clear that $(\mathit{lub}(C^1), \mathit{glb}(C^2)) = \mathit{lub}(C)$. By adapting a proof of [6] (the proof of Proposition 3.10), it can be shown that for any chain $C$ of $\mathcal{L}^{rp}$, $(\mathit{lub}(C^1), \mathit{glb}(C^2)) = (u, v)$ is $\mathcal{A}$-reliable and $\mathcal{A}$-prudent. It follows from Lemma 3 that the operator $St$ is defined on $\mathcal{L}^{rp}$. Then, from Lemma 4 we conclude

**Theorem 1.** *The structure $\langle \mathcal{L}^{rp}, \leq_p \rangle$ is a chain-complete poset that contains the least element $(\bot, \top)$, and $St_{\mathcal{A}}$ is a well-defined, increasing, and $\leq_p$-monotone operator in the poset.*

This completes our theoretical work, which shows that the extended AFT is a sound fixpoint theory; we thus can define:

**Definition 4.** *The least fixpoint of $St_{\mathcal{A}}$ on $\mathcal{L}^{rp}$ is called the* well-founded fixpoint of $\mathcal{A}$ *(and the corresponding semantics the $\mathcal{A}$-WFS) , and the exact fixpoints of $St_{\mathcal{A}}$ on $\mathcal{L}^{rp}$ are called the exact* stable fixpoints *of $\mathcal{A}$ (and the corresponding semantics $\mathcal{A}$-answer set semantics).*

## 4 FOL-Programs

In this section we briefly explore how the extended AFT may be applied to the study of semantics for FOL-programs.

We assume a language of a decidable fragment of first-order logic, denoted $\mathcal{L}_{\Sigma}$, where $\Sigma = \langle F^n; P^n \rangle$, and $F^n$ and $P^n$ are disjoint countable sets of function and predicate symbols, each of which comes with a fixed arity. Constants are 0-ary functions. *Terms* are variables, constants, or functions in the form $f(t_1, ..., t_n)$, where each $t_i$ is a term and $f \in F^n$. *First-order formulas*, or just *formulas*, are defined as usual, so are the notions of *satisfaction*, *model*, and *entailment*.

Let $\Phi_P$ be a finite subset of $P^n$ and $\Phi_C$ a nonempty finite set of constants from $F^n$. An *atom* is of the form $P(t_1, ..., t_n)$ where $P \in \Phi_P$ and each $t_i$ is either a constant from $\Phi_C$ or a variable.

An *FOL-program* is a combined knowledge base $KB = (L, \Pi)$, where $L$ is a first-order theory of $\mathcal{L}_{\Sigma}$ and $\Pi$ a *rule base*, which is a finite set of rules of the form $H \leftarrow A_1, \dots, A_m, not\ B_1, \dots, not\ B_n$, where $H$ and $B_i$ are atoms and $A_i$ are formulas. For any rule $r$, we denote by $hd(r)$ the head of the rule and $body(r)$ its body, and define $pos(r) = \{A_1, ..., A_m\}$ and $neg(r) = \{B_1, ..., B_n\}$.

A *ground instance* of a rule in $\Pi$ is obtained by replacing a free variable with a constant in $\Phi_C$. The process of replacing a rule by all its ground instances is called *grounding*. From now on, we assume $\Pi$ is already grounded. When we refer to an atom/literal/formula, by default we mean it is a ground one.

Given an FOL-program $KB = (L, \Pi)$, the *Herbrand base* of $\Pi$, denoted $HB_{\Pi}$, is the set of all ground atoms $P(t_1, ..., t_n)$, where $P \in \Phi_P$ and $t_i \in \Phi_C$.

For ease of presentation, we assume that $\Phi_P$ only contains predicate symbols that occur in $\Pi$, and it contains at least all predicate symbols that occur in $\Pi$ but not in $L$. Under this assumption, no predicate symbol that appears in $L$ but not in $\Pi$ may be in the underlying Herbrand base, and every predicate symbol appearing in $\Pi$ but not in $L$ is necessarily in the underlying Herbrand base. Recall that answer sets and WFS are only concerned with atoms in the underlying Herbrand base.

Any subset $I \subseteq HB_{\Pi}$ is called an *interpretation* of $\Pi$. If $I$ is a set of (ground) atoms, we define $\bar{I} = HB_{\Pi} \backslash I$, and $\neg.I = \{\neg A \mid A \in I\}$.

Given lattice $\langle 2^{HB_{\Pi}}, \subseteq \rangle$, the induced bilattice is $\langle (2^{HB_{\Pi}})^2, \subseteq_p \rangle$. A consistent pair $(I, J)$ in $(2^{HB_{\Pi}})^2$ represents a *partial interpretation* $I \cup \neg.\bar{J}$. Let $(I, J) \in (2^{HB_{\Pi}})^2$ and $L$ a first-order theory. We say that $(I, J)$ is consistent with $L$ if $L \cup I \cup \neg.\bar{J}$ is consistent, and $(I', J')$ is a *consistent extension of* $(I, J)$ if $I \subseteq I' \subseteq J' \subseteq J$ (if $(I, J)$ is inconsistent, such $(I', J')$ does not exist).

**Definition 5.** *Let $KB = (L, \Pi)$ be an FOL-program, $(I, J) \in (2^{HB_{\Pi}})^2$, and $\phi$ a literal. We define two entailment relations, which extend to conjunctions of literals.*

- $(I, J) \models_L \phi$ *iff, if $\phi$ is an atom $A$ then $A \in I$, if $\phi$ is a negative literal $not\ A$ then $A \notin I$,*[6] *and if $\phi$ is an FOL-formula then $L \cup I \cup \neg.\bar{J} \models \phi$.*
- $(I, J) \Vdash_L \phi$ *iff for all consistent extensions $(I', J')$ of $(I, J)$, $(I', J') \models_L \phi$.*

**Operator to be Approximated:** Let $KB = (L, \Pi)$ be an FOL-program. Define an operator $\mathcal{K}_{KB}\colon 2^{HB_{\Pi}} \rightarrow 2^{HB_{\Pi}}$ as follows: for any $I \in 2^{HB_{\Pi}}$

$$\mathcal{K}_{KB}(I) = \{hd(r) \mid r \in \Pi, (I, I) \models_L body(r)\} \cup \{A \in HB_{\Pi} \mid (I, I) \models_L A\}$$

[6] Default negation here is evaluated independently of $L$, which has been called *local closed world reasoning* [11].

This operator is essentially the immediate consequence operator augmented by *direct positive consequences*. Elements in $2^{HB_\Pi}$ are (2-valued) interpretations, which are inherently weak in representing inconsistency. In the extended AFT, it is the inconsistent pairs that tie up this loose end, by explicitly representing negative information.

In the definitions below, we assume an FOL-program $KB = (L, \Pi)$, $(I, J) \in (2^{HB_\Pi})^2$, and $v \in 2^{HB_\Pi}$ such that $J \subseteq v$.

**Definition 6.** (Operator $\Phi_{KB,v}$: Standard Semantics) *For all $H \in HB_\Pi$,*

- *$H \in \Phi^1_{KB,v}(I, J)$ iff one of the following holds*
  *(a)* $(I, v) \models_L H$.
  *(b)* $\exists r \in \Pi$ *with* $hd(r) = H$, *s.t.* $\forall \phi \in body(r)$, $(I, J) \Vdash_L \phi$.
- *$H \in \Phi^2_{KB,v}(I, J)$ iff $(I, v) \not\models_L \neg H$ and one of the following holds*
  *(a)* $\exists I', J'(I \subseteq I' \subseteq J' \subseteq J)$, $(I', J' \cup \bar{J}) \models_L H$.
  *(b)* $\exists r \in \Pi$ *with* $hd(r) = H$, *s.t.* $\forall \phi \in body(r)$, $\exists I', J'(I \subseteq I' \subseteq J' \subseteq J)$, $(I', J' \cup \bar{J}) \models_L \phi$.

Operator $\Phi^1_{KB,v}$ computes atoms that must be true, either due to *directly entailed* w.r.t. $L$ (part (a)), or *persistently derivable* (part (b)). Operator $\Phi^2_{KB,v}$ on the other hand computes possibly true atoms, under the condition that their complements are not entailed by $(I, v)$, that are either *potentially entailed* (part (a)), or *possibly derivable* (part (b), in which each body literal may be derived from a different $(I', J')$).

Note that the pair $(I, J)$ in $\Phi^2_{KB,v}(I, J)$ serves as an interval $[I, J]$ in the sense that any atom in it may be assigned arbitrarily, as expressed in $(I', J' \cup \bar{J})$, as a partial interpretation that extends $I$, in that the atoms in $I'$ are assigned to true and those in $J$ but not in $J'$ are assigned to false.

If $(I, v)$ is inconsistent with $L$, then the result is $(HB_\Pi, \emptyset)$.

**Lemma 5.** *$\Phi_{KB}$ is an approximator of $\mathcal{K}_{KB}$.*

*Example 3.* Consider Example 2 again: $KB = (L, \Pi)$ where $L = \{\forall x\, C(x) \supset (A(x) \vee D(x))\}$ and $\Pi = \{A(a) \leftarrow A(a).\ B(a) \leftarrow not\ A(a).\ D(a) \leftarrow not\ B(a).\ C(a) \leftarrow not\ C'(a).\ C'(a) \leftarrow not\ C(a).\}$ The stable revision operator, $St_{\Phi_{KB}}$ in this case, generates the below sequence (recall that we write $a$ for $A(a)$, $b$ for $B(a)$, and so on):

$$(\emptyset, HB_\Pi) \Rightarrow (\emptyset, \{c, c', b, d\}) \Rightarrow (\{b\}, \{c, c', b, d\}) \Rightarrow (\{b\}, \{c, c', b\}) \Rightarrow (\{b\}, \{c', b\}) \Rightarrow (\{c', b\}, \{c', b\})$$

$A(a)$ is false by the second pair, which leads to the derivation of $B(a)$ and then to the inference that $D(a)$ is false, in the next two steps. Next, as $\neg C(a)$ is entailed by $L$, $C(a)$ is no longer possible true. Finally, $C'(a)$ is derived to be true.

The above example shows how "under estimate" is corrected under the extended AFT here, which improves the work of [2]. In the next example, we illustrate how inconsistency is handled.

*Example 4.* Let $KB = (\{\neg A(a)\}, \Pi)$ where $\Pi = \{A(a) \leftarrow not\ B(a);\ B(a) \leftarrow B(a);\ C(a) \leftarrow\}$. Let $\Phi_P = \{A, B, C\}$ and $\Phi_C = \{a\}$. The well-founded fixpoint of $\Phi_{KB}$ is computed by the stable revision operator $St_{\Phi_{KB}}$ as follows:

$$(\emptyset, HB_\Pi) \Rightarrow (\{C(a)\}, \{C(a)\}) \Rightarrow (HB_\Pi, \{C(a)\}) \Rightarrow (HB_\Pi, \emptyset\})$$

From the second pair, we get $lfp(\Phi^1_{KB,v'}(\cdot, \{C(a)\})) = HB_\Pi$, due to inconsistency from the derivation of $A(a)$. Let us denote the third pair by $(u, v)$. Because $\Phi^2_{KB,v}(u, u) = \emptyset$ (thus $\not\supseteq u$), the second case in Def. 3 is triggered, and leads to $lfp(\Phi^2_{KB,v}(u, \cdot)) = \emptyset$, which together with $lfp(\Phi^1_{KB,v}(\cdot, v))$ yields the last pair as the fixpoint.

The well-founded and answer set semantics based on the operator $\Phi_{KB}$ are called *standard*, because they are generalizations of the WFS and answer set semantics for normal logic programs.

**Theorem 2.** *Let $KB = (\emptyset, \Pi)$ be a normal program, i.e., $L = \emptyset$ and $\Pi$ is a set of* normal rules. *Then, the $\Phi$-WFS of KB coincides with the WFS of $\Pi$ [19], and $\Phi$-answer set semantics coincides with the standard stable model semantics of $\Pi$ [10].*

## 5 Related Work and Future Directions

AFT has applied to DL-programs [8], which can be represented by HEX-programs [1] and aggregate programs [14], where an approximator can be defined so that the well-founded fixpoint defines the WFS [9] and the exact stable fixpoints define the well-supported answer set semantics [16]. In [12], a well-founded semantics for combing rules with DLs is defined. In both approaches above, syntactic restrictions are imposed so that the least fixpoint is always constructed over sets of consistent literals.

Our work here is built on top of the approach in [2]. Because there is no explicit representation of entailed negative information in approximators, the phenomenon of over estimate and under estimate does arise in the approach of [2].

In our extended AFT, inconsistency handling relies on how an approximator is defined. So far, the approximators defined allow trivialization to take place, but practically one may define approximators so that non-trivialization is supported. E.g., in Example 4, one may define the approximator $\Phi_{KB}$ in a way that $(\{C(a)\}, \{C(a)\})$ is stable revised to $(\{A(a), C(a)\}, \{C(a)\})$, where $A(a)$ being true but not possible true indicates where inconsistency initially occurred. Further investigation in this direction is needed.

Recently, the theory of *grounded fixpoints* has generated some interest [4]. Again, the work in general does not treat inconsistent pairs on bilattices. It is interesting to investigate how the approach proposed in this paper may be applied. In addition, the notion of *unfounded atoms* by non-derivation of rules with arbitrary formulas in bodies [3, 4] is interesting. In general, the relationship between unfoundedness and stable revision for various classes of programs, including FOL-programs, requires a further study.

In this paper, we only provided an initial attempt to show applications of the extended AFT. We will look into other applications where a separation of established negation and default negation is desirable. Tightly coupled multi-context systems is a potential target.

## References


1. Christian Antic, Thomas Eiter, and Michael Fink. Hex semantics via approximation fixpoint theory. In *Proc. LPNMR*, pages 102–115, 2013.
2. Yi Bi, Jia-Huai You, and Zhiyong Feng. A generalization of approximation fixpoint theory and application. In *Proc. 8th Int'l Conference on Web Reasoning and Rule Systems*, pages 45–59, 2014.
3. Bart Bogaerts, Joost Vennekens, and Marc Denecker. Grounded fixpoints. In *Proc. AAAI'15*, pages 1453–1459, 2015.
4. Bart Bogaerts, Joost Vennekens, and Marc Denecker. Grounded fixpoints and their applications in knowledge representation. *Artif. Intell.*, 224:51–71, 2015.
5. Marc Denecker, Victor W. Marek, and Miroslaw Truszczynski. Uniform semantic treatment of default and autoepistemic logics. *Artif. Intell.*, 143(1):79–122, 2003.
6. Marc Denecker, Victor W. Marek, and Miroslaw Truszczynski. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation*, 192(1):84–121, 2004.
7. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.
8. Thomas Eiter, Giovambattista Ianni, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. *Artif. Intell.*, 172(12-13):1495–1539, 2008.
9. Thomas Eiter, Thomas Lukasiewicz, Giovambattista Ianni, and Roman Schindlauer. Well-founded semantics for description logic programs in the semantic web. *ACM Transactions on Computational Logic*, 12(2), 2011.
10. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proc. ICLP'88*, pages 1070–1080, 1988.
11. Matthias Knorr, José Júlio Alferes, and Pascal Hitzler. Local closed world reasoning with description logics under the well-founded semantics. *Artif. Intell.*, 175(9-10):1528–1554, 2011.
12. Thomas Lukasiewicz. A novel combination of answer set programming with description logics for the semantic web. *IEEE TKDE*, 22(11):1577–1592, 2010.
13. Boris Motik and Riccardo Rosati. Reconciling description logics and rules. *J. ACM*, 57(5):1–62, 2010.
14. Maurice Bruynooghe Nikolay Pelov, Marc Denecker. Well-founded and stable semantics of logic programs with aggregates. *Theory and Practice of Logic Programming*, 7:301–353, 2007.
15. Riccardo Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In *Proc. KR'06*, pages 68–78, 2006.
16. Yi-Dong Shen and Kewen Wang. Extending logic programs with description logic expressions for the semantic web. In *Proc. Int'l Semantic Web Conference*, pages 633–648, 2011.
17. Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.
18. Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:2:285–309, 1955.
19. A. van Gelder, K. Ross, and J. Schlipf. The well-founded semantics for general logic programs. *J. ACM*, 38(3):620–650, 1991.
20. Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. FO(ID) as an extension of DL with rules. *Ann. Math. Artif. Intell.*, 58(1-2):85–115, 2010.
21. Qian Yang, Jia-Huai You, and Zhiyong Feng. Integrating rules and description logics by circumscription. In *Proc. AAAI'11*, 2011.