Partial Grounded Fixpoints

Bart Bogaerts, Joost Vennekens, Marc Denecker
Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
{bart.bogaerts,joost.vennekens,marc.denecker}@cs.kuleuven.be

Abstract

Approximation fixpoint theory (AFT) is an algebraical study of fixpoints of lattice operators. Recently, AFT was extended with the notion of a grounded fixpoint. This type of fixpoint formalises common intuitions from various application domains of AFT, including logic programming, default logic, autoepistemic logic and abstract argumentation frameworks.

The study of groundedness was limited to exact lattice points; in this paper, we extend it to the bilattice: for an approximator A of O, we define A-groundedness. We show that all partial A-stable fixpoints are A-grounded and that the A-well-founded fixpoint is uniquely characterised as the least precise A-grounded fixpoint. We apply our theory to logic programming and study complexity.

1 Introduction

Approximation fixpoint theory (AFT) is an algebraical unifying study of semantics of nonmonotonic logics. Given a lattice operator and a so-called approximating bilattice operator, Denecker, Marek and Truszczyński (henceforth DMT) [2000] defined several types of fixpoints. They showed that all of the main semantics of logic programming are induced by AFT, using Fitting's partial immediate consequence operator as an approximator of the two-valued immediate consequence operator. They identified approximating operators for default logic (DL) [Reiter, 1980] and autoepistemic logic (AEL) [Moore, 1985] and showed that AFT induces all main and some new semantics in these fields [Denecker *et al.*, 2003].

Recently, AFT was used in a broader context. It was used to study properties such as modularity and predicate introduction in a unifying framework by Vennekens *et al.* [2007; 2006]. Strass [2013] showed that many semantics for Dung's argumentation frameworks (AFs) [Dung, 1995] and abstract dialectical frameworks (ADFs) [Brewka and Woltran, 2010] can be retrieved by application of AFT. In this study, he also extends AFT with new types of fixpoints. Bi *et al.* [2014] extended AFT to account for inconsistencies and used this to integrate logic programs with description logics. Bogaerts *et al.* [2014] defined the causal logic FO(C) based on AFT.

Bogaerts *et al.* [2015] extended AFT with a concept called *groundedness* which is a surprisingly simple formalisation of intuitions that have existed in different research domains for years. They showed that grounded fixpoints are an intuitive concept, closely related to exact (two-valued) stable fixpoints. In the context of logic programming, grounded fixpoints can be characterised using a generalised notion of unfounded set.

Grounded fixpoints are lattice elements; in this work, we generalise them to points in the bilattice, while still maintaining the elegance and desirable properties of groundedness. For the case of logic programming, this generalisation boils down to extending groundedness to partial (or three-valued) interpretations. There are several reasons why it is important to generalise a notion as groundedness to a partial context.

The first, and probably most obvious one, is that sometimes partial fixpoints are a central topic of study themselves. As an example, in AFs and ADFs, most of the studied semantics are three-valued.

A second reason is that studying concepts in the bilattice (i.e., in a partial context) is more general than studying them only in the original lattice. For example, many of our results are generalisations of results from Bogaerts *et al.* [2015]; sometimes, these generalisations even fill holes in their theory. E.g., while Bogaerts *et al.* [2015] showed that *exact* A-stable fixpoints are grounded, we show that *all* A-stable fixpoints are A-grounded. Also, while Bogaerts *et al.* [2015] showed that, if the well-founded fixpoint is exact, then it is grounded, we show that the A-well-founded fixpoint is *always* A-grounded. This last result illustrates that the well-founded semantics does a great job at avoiding ungrounded models: the well-founded model is not *just* some grounded model, it is the *least precise* grounded model.

A third reason is that even in domains where we are only interested in two-valued models, partial models can be useful, e.g., as an analysis tool. For example, in answer set programming [Marek and Truszczyński, 1999] almost all specifications are designed for two-valued stable model semantics. Errors in such specifications may lead to logic programs without stable models; they are hard to debug. Partial stable models [Przymusinski, 1991] can help the debugging process as they can sometimes pinpoint the reason for inconsistency. As a trivial example, let Σ be a vocabulary and p a symbol not in Σ. Augmenting any logic program over Σ with the rule

$$p \leftarrow \neg p$$

yields a logic program without stable models. Debugging tools can use the observation that p is undefined in every partial stable model to identify the "mistake" in the above program, namely the extra rule. The technique of debugging knowledge bases based on partial stable models is used for example in the XNMR system [Castro and Warren, 2001]. Partial stable models are not only used for debugging. Real-life databases often contain (local) inconsistencies. The partial stable model semantics for deductive databases is more *robust* than the two-valued stable semantics in this case. It only assigns the value *unknown* to atoms in an inconsistent part of the database, hence it still facilitates inference on the rest of the database [Seipel *et al.*, 1997; Eiter *et al.*, 1997].

Summarised, the main contributions of this paper are as follows.

1. We extend the notion of groundedness to points in the bilattice.
2. We study the relationship between partial grounded fixpoints and the other fixpoints studied in AFT. An important result in this context is that we find a new characterisation of the well-founded fixpoint based on groundedness.
3. We study how partial grounded fixpoints depend on the choice of an approximator.
4. We apply our theory to logic programming.
5. We study complexity of both credulous and sceptical reasoning of partial grounded model semantics in logic programming.

2 Preliminaries

In this section, we recall the basics of AFT. Our presentation follows the preliminaries section from Bogaerts *et al.* [2015].

Lattices and Operators A *complete lattice* $\langle L, \leq \rangle$ is a set L equipped with a partial order $\leq$, such that every set $S \subseteq L$ has both a least upper bound and a greatest lower bound, denoted $\mathrm{lub}(S)$ and $\mathrm{glb}(S)$ respectively. A complete lattice has a least element $\bot$ and a greatest element $\top$. As custom, we also use the notations $\bigwedge S = \mathrm{glb}(S)$, $x \wedge y = \mathrm{glb}(\{x, y\})$, $\bigvee S = \mathrm{lub}(S)$ and $x \vee y = \mathrm{lub}(\{x, y\})$.

An operator $O : L \to L$ is *monotone* if $x \leq y$ implies that $O(x) \leq O(y)$. An element $x \in L$ is a *prefixpoint*, a *fixpoint*, a *postfixpoint* if $O(x) \leq x$, respectively $O(x) = x$, $x \leq O(x)$. Every monotone operator O in a complete lattice has a least fixpoint, denoted $\mathrm{lfp}(O)$, which is also O's least prefixpoint.

Approximation Fixpoint Theory Given a lattice L, AFT makes uses of the bilattice L^2. We define *projections* for pairs as usual: $(x, y)_1 = x$ and $(x, y)_2 = y$. Pairs $(x, y) \in L^2$ are used to approximate all elements in the interval $[x, y] = \{z \mid x \leq z \wedge z \leq y\}$. We call $(x, y) \in L^2$ *consistent* if $x \leq y$, that is, if $[x, y]$ is non-empty. We use L^c to denote the set of consistent elements. Elements $(x, x) \in L^c$ are called *exact*. We sometimes abuse notation and use the tuple (x, y) and the interval $[x, y]$ interchangeably. The *precision order* on L^2 is defined as $(x, y) \leq_p (u, v)$ if $x \leq u$ and $v \leq y$. If (u, v) is consistent, this means that (x, y) approximates all elements approximated by (u, v), or in other words that $[u, v] \subseteq [x, y]$. If L is a complete lattice, then so is $\langle L^2, \leq_p \rangle$.

AFT studies fixpoints of lattice operators $O : L \to L$ through operators approximating O. An operator $A : L^2 \to L^2$ is an *approximator* of O if it is $\leq_p$-monotone, and has the property that for all x, $O(x) \in [x', y']$, where $(x', y') = A(x, x)$. Approximators are internal in L^c (i.e., map L^c into L^c). As usual, we restrict our attention to *symmetric* approximators: approximators A such that for all x and y, $A(x, y)_1 = A(y, x)_2$. DMT [2004] showed that the consistent fixpoints of interest (supported, stable, well-founded) are uniquely determined by an approximator's restriction to L^c, hence, sometimes we only define approximators on L^c.

AFT studies fixpoints of O using fixpoints of A.

- The *A-Kripke-Kleene fixpoint* is the $\leq_p$-least fixpoint of A and has the property that it approximates all fixpoints of O.
- A *partial A-stable fixpoint* is a pair (x, y) such that $x = \mathrm{lfp}(A(\cdot, y)_1)$ and $y = \mathrm{lfp}(A(x, \cdot)_2)$, where $A(\cdot, y)_1$ denotes the operator $L \to L : x \mapsto A(x, y)_1$ and analogously for $A(x, \cdot)_2$.
- The *A-well-founded fixpoint* is the least precise partial A-stable fixpoint.
- An *A-stable fixpoint* of O is a fixpoint x of O such that (x, x) is a partial A-stable fixpoint.

The A-Kripke-Kleene fixpoint of O can be constructed by iteratively applying A, starting from $(\bot, \top)$. For the A-well-founded fixpoint, a similar constructive characterisation has been worked out by Denecker and Vennekens [2007].

Definition 2.1. An *A-refinement* of (x, y) is a pair $(x', y') \in L^2$ satisfying one of the following two conditions:

- $(x, y) \leq_p (x', y') \leq_p A(x, y)$, or
- $x' = x$ and $A(x, y')_2 \leq y' \leq y$.

An A-refinement is *strict* if $(x, y) \neq (x', y')$.

Definition 2.2. A *well-founded induction* of A is a sequence $(x_i, y_i)_{i \leq \beta}$ with β an ordinal such that

- $(x_0, y_0) = (\bot, \top)$;
- (x_{i+1}, y_{i+1}) is an A-refinement of (x_i, y_i), for all $i < \beta$;
- $(x_\lambda, y_\lambda) = \mathrm{lub}_{\leq_p} \{(x_i, y_i) \mid i < \lambda\}$ for each limit ordinal $\lambda \leq \beta$.

A well-founded induction is *terminal* if its limit (x_β, y_β) has no strict A-refinements.

For an approximator A, there are many different terminal well-founded inductions of A. Denecker and Vennekens [2007] showed that they all have the same limit, which equals the A-well-founded fixpoint of O. If A is symmetric, the A-well-founded fixpoint of O (and in fact, every tuple in a well-founded induction of A) is consistent.

In general, a lattice operator $O : L \to L$ has a family of approximators of different precision. For two approximators A and B of O, we say that $A \leq_p B$ if $A(x, y) \leq_p B(x, y)$

for all $(x,y) \in L^c$. In this case, all A-stable fixpoints are B-stable fixpoints, and the B-well-founded fixpoint is more precise than the A-well-founded fixpoint. DMT [2004] showed that there exists a most precise approximator, U_O, called the ultimate approximator of O. This operator is defined by $U_O : L^c \to L^c : (x,y) \mapsto (\bigwedge O([x,y]), \bigvee O([x,y]))$, where $O([x,y]) = \{O(z) \mid z \in [x,y]\}$.

Bogaerts *et al.* [2015] called a point $x \in L$ *grounded* for O if for each $v \in L$ such that $O(v \wedge x) \leq v$, it holds that $x \leq v$. They explained the intuition underlying this concept under the assumption that the elements of L are sets of "facts" of some kind and the $\leq$ relation is the subset relation between such sets: in this case, a point x is grounded if it contains only facts that are sanctioned by the operator O, in the sense that if we remove them from x, then the operator will add at least one of them again.

AFT and Logic Programming We restrict our attention to propositional logic programs with arbitrary propositional formulas in rule bodies. AFT has been applied in a much broader context [Denecker *et al.*, 2012; Pelov *et al.*, 2007; Antic *et al.*, 2013] and our results extend there as well.

Let Σ be an alphabet, i.e., a collection of symbols which are called *atoms*. A logic program $\mathcal{P}$ is a set of *rules* r of the form $h \leftarrow \varphi$, where h is an atom called the *head* of r, denoted $head(r)$, and φ is a propositional formula called the *body* of r, denoted $body(r)$. An interpretation I of Σ is a subset of Σ. The set of interpretations 2^Σ forms a lattice equipped with the order $\subseteq$. The truth value ($\mathbf{t}$ or $\mathbf{f}$) of a propositional formula φ in a structure I, denoted φ^I, is defined as usual. With a logic program $\mathcal{P}$, we associate an immediate consequence operator [van Emden and Kowalski, 1976] $T_\mathcal{P}$ that maps a structure I to

$$\{p \mid \exists r \in \mathcal{P} : head(r) = p \wedge body(r)^I = \mathbf{t}\}.$$

Elements of the bilattice $\left(2^\Sigma\right)^2$ are four-valued interpretations, pairs $\mathcal{I} = (I_1, I_2)$ of interpretations. The pair (I_1, I_2) approximates all interpretations I' with $I_1 \subseteq I' \subseteq I_2$. We are mostly concerned with three-valued (or, partial) interpretations: tuples $\mathcal{I} = (I_1, I_2)$ with $I_1 \subseteq I_2$. For such an interpretation, the atoms in I_1 are *true* ($\mathbf{t}$) in $\mathcal{I}$, the atoms in $I_2 \setminus I_1$ are *unknown* ($\mathbf{u}$) in $\mathcal{I}$ and the other atoms are *false* ($\mathbf{f}$) in $\mathcal{I}$. If $\mathcal{I}$ is a three-valued interpretation, and φ a formula, we write $\varphi^\mathcal{I}$ for the standard three-valued valuation based on Kleene's truth tables [Kleene, 1938]. We identify interpretation I with the partial interpretation (I, I). If $\mathcal{I} = (I_1, I_2)$ is a (partial) interpretation, and $U \subseteq \Sigma$, we write $\mathcal{I}[U : \mathbf{f}]$ (respectively $\mathcal{I}[U : \mathbf{t}]$) for the (partial) interpretation that equals $\mathcal{I}$ on all elements not in U and that interprets all elements in U as $\mathbf{f}$ (respectively $\mathbf{t}$), i.e., the interpretation $(I_1 \setminus U, I_2 \setminus U)$ (respectively $(I_1 \cup U, I_2 \cup U)$).

Several approximators have been defined for logic programs. The most common is Fitting's immediate consequence operator $\Psi_\mathcal{P}$ [Fitting, 2002], a direct generalisation of $T_\mathcal{P}$ to partial interpretations defined by

$$\Psi_\mathcal{P}(\mathcal{I})_1 = \{p \mid \exists r \in \mathcal{P} : head(r) = p \wedge body(r)^\mathcal{I} = \mathbf{t}\}$$
$$\Psi_\mathcal{P}(\mathcal{I})_2 = \{p \mid \exists r \in \mathcal{P} : head(r) = p \wedge body(r)^\mathcal{I} \neq \mathbf{f}\}$$

Denecker *et al.* [2000] showed that the $\Psi_\mathcal{P}$-well-founded fixpoint is the well-founded model of $\mathcal{P}$ [Van Gelder *et al.*, 1991] and that $\Psi_\mathcal{P}$-stable fixpoints are exactly the stable models of $\mathcal{P}$ [Gelfond and Lifschitz, 1988].

In the context of logic programming, Bogaerts *et al.* [2015] defined that a set $U \subseteq \Sigma$ is a *2-unfounded set* of $\mathcal{P}$ with respect to interpretation I if $T_\mathcal{P}(I[U : \mathbf{f}]) \cap U = \emptyset$. Intuitively, a 2 unfounded set is a set of atoms such that if you make them false (starting from I), they stay false (when applying $T_\mathcal{P}$). They showed that this definition is a slight variant of the notion of unfounded set defined by Van Gelder *et al.* [1991]. Furthermore, they generalised[1] this definition to partial interpretations, resulting in a definition which is equivalent with the definition by Van Gelder *et al.* [1991] for all interpretations used in the well-founded model construction. Bogaerts *et al.* [2015] showed that an interpretation I is grounded for $T_\mathcal{P}$ if and only if I does not contain any atoms that belong to a 2-unfounded set U of P with respect to I.

3 Partial Grounded Fixpoints

We start by giving the central definition of this text, namely the notion of A-groundedness.

Definition 3.1. Let A be an approximator of O. A point $(x,y) \in L^c$ is *A-grounded* if for every $v \in L$ with $A(x \wedge v, y \wedge v)_2 \leq v$, also $y \leq v$.

The definition of this concept is a direct generalisation of groundedness for points $x \in L$. It is based on the same intuitions, but applied in a more general context. We again explain the intuitions under the assumption that the elements of L are sets of "facts" and the $\leq$ relation is the subset relation between such sets. In this case, a point $(x,y) \in L^c$ represents a partial set of "facts": the elements of x are the facts that *certainly* belong to the partial set, the elements in y are those that *possibly* belong to the set. Thus, $y \setminus x$ are the *unknown* elements and the complement of y are those that do not belong to the set. In this case, a point (x,y) is A-grounded if all its possible facts (those in y) are sanctioned by the operator A, in the sense that if we remove some elements from the partial set, then A will make at least one of them possible again. The above definition captures this idea, by using a set $v \in L$ to remove all elements not in v from (x,y): the point $(x \wedge v, y \wedge v)$ does not contain facts in the complement of v; on all other facts, $(x \wedge v, y \wedge v)$ equals (x,y). In an A-grounded point (x,y) removing elements must result in a state where at least one of them is re-derived to be possible: it must hold that $A(x \wedge v, y \wedge v)_2 \not\leq v$. Stated differently, if the removal of elements from (x,y) is not contradicted by A, i.e., applying A results in a state where these elements are still false, then these elements cannot be part of the grounded point ($y \leq v$). In what follows, we study properties of A-grounded (fix)points, including their relation to other fixpoints studied in AFT. A first observation is that for exact points, i.e., bilattice points of the form (x,x), the choice of the approximator A does not matter and our notion of groundedness coincides with the previously defined.

[1] We come back to this in Section 4 and show that there are in fact many ways to extend this definition to partial interpretations.

Proposition 3.2. *If A is a symmetric approximator of O, then x is grounded for O if and only if (x, x) is A-grounded.*

Proof. Trivial. Follows immediately from the fact that $A(x, x)_2 = O(x)$ for all $x \in L$ if A is symmetric. $\square$

A second observation is that all partial A-stable fixpoints are A-grounded. This is a generalisation of Corollary 4.5 from Bogaerts *et al.* [2015] which showed that all exact stable fixpoints are grounded.

Proposition 3.3. *All consistent (partial) A-stable fixpoints are A-grounded.*

Proof. Suppose $(x, y) \in L^c$ is an A-stable fixpoint, i.e., $x = \mathrm{lfp}(A(\cdot, y)_1)$ and $y = \mathrm{lfp}(A(x, \cdot)_2)$. Also assume that for some v, $A(x \wedge v, y \wedge v)_2 \leq v$. We show that $y \leq v$. Since A is $\leq_p$-monotone and $y \wedge v \leq y$, it holds that

$$A(x, y \wedge v)_2 \leq A(x, y)_2 = y.$$

Analogously, since A is $\leq_p$-monotone and $x \wedge v \leq x$, it holds that

$$A(x, y \wedge v)_2 \leq A(x \wedge v, y \wedge v)_2 \leq v.$$

Combining these two observations, we find that

$$A(x, y \wedge v)_2 \leq y \wedge v,$$

i.e., that $y \wedge v$ is a prefixpoint of the monotone operator $A(x, \cdot)_2$. Since y is the least prefixpoint of this operator, it must hold that $y \leq y \wedge v$, thus also that $y \leq v$, which we needed to show. $\square$

Example 3.4. The converse of Proposition 3.3 does not hold. Consider the logic program $\mathcal{P}$

$$\left\{ \; p \leftarrow p \vee \neg p. \; \right\}$$

We claim that every bilattice point is $\Psi_{\mathcal{P}}$-grounded in this case. Indeed, if $(x, y) \in L^2$, then for $v = \{p\}$, $y \leq v$ is trivially true, for $v = \emptyset$, it holds that

$$\Psi_{\mathcal{P}}(x \wedge v, y \wedge v)_2 = \Psi_{\mathcal{P}}(\emptyset, \emptyset)_2 = \{p\} \not\leq \emptyset = v.$$

Hence, all points in the bilattice are $\Psi_{\mathcal{P}}$-grounded. Thus, $(\{p\}, \{p\})$ is a $\Psi_{\mathcal{P}}$-grounded fixpoint which is not $\Psi_{\mathcal{P}}$-stable.

A third observation is that the A-well-founded fixpoint is less precise than any A-grounded fixpoint. This property generalises both the fact that the A-well-founded fixpoint approximates all *exact* grounded fixpoints of O (Theorem 4.6 from Bogaerts *et al.* [2015]) and the fact that the A-well-founded fixpoint is less precise than any partial A-stable fixpoint (Theorem 23 (2) from DMT [2000]).

Theorem 3.5. *The well-founded fixpoint (u, v) of a symmetric approximator A of O is less precise than any A-grounded fixpoint.*

Before we prove this theorem, we show that the second type of refinements in a well-founded induction eliminates only non-grounded bilattice points.

Lemma 3.6. *Let (a, b) and (a, b') be elements of L^c such that $A(a, b')_2 \leq b' \leq b$. Then for every b'' with $b' \lneq b'' \leq b$, (a, b'') is ungrounded.*

Proof. For every such b'' it holds that $A(a \wedge b', b'' \wedge b')_2 = A(a, b')_2 \leq b'$ while $b'' \not\leq b'$. $\square$

Proof of Theorem 3.5. Let $(a_i, b_i)_{i \leq \beta}$ be a well-founded induction of A and let (x, y) be an A-grounded fixpoint. We show by induction that for every $i \leq \beta$, $a_i \leq x$ and $y \leq b_i$. The results trivially holds for $i = 0$ since $(a_0, b_0) = (\bot, \top)$. It is also clear that the property is preserved in limit ordinals. Hence, all we need to show is that the property is preserved by A-refinements. Suppose (a', b') is an A-refinement of (a, b) and that $(a, b) \leq_p (x, y)$. We show that $(a', b') \leq_p (x, y)$. We distinguish two cases. If $(a, b) \leq_p (a', b') \leq_p A(a, b)$, then since A is $\leq_p$-monotone and $(a, b) \leq_p (x, y)$, it holds that $A(a, b) \leq_p A(x, y) = (x, y)$, hence also $(a', b') \leq_p (x, y)$. For the second type of refinements, it follows from Lemma 3.6 that only ungrounded bilattice points are removed. $\square$

It is well-known that the A-well-founded fixpoint is characterised as the least precise A-stable fixpoint. The previous theorem provides us with a similar characterisation of the well-founded fixpoints in terms of groundedness. This is an important result, as it again supports the claim that many of the existing semantics are designed with the intuitions of groundedness in mind: we establish a very tight link between the well-founded fixpoints and grounded fixpoints.

Corollary 3.7. *The well-founded fixpoint of a symmetric approximator A of O is the least precise A-grounded fixpoint.*

Proof. The well-founded fixpoint of A is consistent and A-stable, hence Proposition 3.3 shows that it is A-grounded as well. Theorem 3.5 shows that it is less precise than any A-grounded fixpoint, hence the result follows. $\square$

An operator can have multiple approximators. In Proposition 3.2, we already showed that the choice of approximator A does not affect *exact* A-grounded fixpoints: these are exactly the fixpoints that are grounded for O. The question still remains how groundedness of other points in the bilattice is influenced by such a choice. The next proposition answers this question.

Proposition 3.8. *If A and B are approximators of O and $A \leq_p B$, then all consistent B-grounded points are also A-grounded.*

Proof. Suppose $(x, y) \in L^c$ is B-grounded; we show that it is also A grounded. Assume that for some v it holds that $A(x \wedge v, y \wedge v)_2 \leq v$. Since $A \leq_p B$, it holds that

$$B(x \wedge v, y \wedge v)_2 \leq A(x \wedge v, y \wedge v)_2 \leq v.$$

Since (x, y) is B-grounded, we find that $y \leq v$. $\square$

Example 3.9. The choice of an approximator can really make a difference for the notion of A-groundedness, i.e., not all A-grounded points are B-grounded in Proposition 3.8. Consider the logic program $\mathcal{P}$

$$\left\{ \begin{array}{l} p \leftarrow \neg p. \\ q \leftarrow q. \end{array} \right\}$$

Then $(\{q\}, \{p, q\})$ is not $\Psi_\mathcal{P}$-grounded since

$$\begin{aligned}\Psi_\mathcal{P}(\{q\} \wedge \{p\}, \{p, q\} \wedge \{p\})_2 &= \Psi_\mathcal{P}(\emptyset, \{p\})_2\\ &= (\emptyset, \{p\})_2 = \{p\},\end{aligned}$$

while $\{p, q\} \not\leq \{p\}$.

However, let A be the trivial approximator of $T_\mathcal{P}$, i.e., the approximator such that $A(x, x) = (T_\mathcal{P}(x), T_\mathcal{P}(x))$ for every $x \in L$ and $A(x, y) = (\bot, \top)$ for $(x, y) \in L^c$ with $x \neq y$. We claim that $(\{q\}, \{p, q\})$ is A-grounded. We prove this claim by showing that all $v \in L$ satisfy the condition from Definition 3.1. For $v = \top$ the condition $y \leq \top$ is trivially satisfied. For $v = \{q\}$, we find

$$A(\{q\} \wedge \{q\}, \{p, q\} \wedge \{q\})_2 = \{p, q\} \not\leq \{q\}.$$

For $v = \{p\}$, we find

$$A(\{q\} \wedge \{p\}, \{p, q\} \wedge \{p\})_2 = \top \not\leq \{p\}$$

since $(\{q\} \wedge \{p\}, \{p, q\} \wedge \{p\})$ is not exact. For $v = \emptyset$, we find that

$$A(\{q\} \wedge \emptyset, \{p, q\} \wedge \emptyset)_2 = \{p\} \not\leq \emptyset.$$

This indeed proves our claim.

4 Partial Grounded Fixpoints in Logic Programming

In this section, we apply our theory to logic programming. More concretely we extend the observation that in logic programming, groundedness is closely related to unfounded sets [Bogaerts *et al.*, 2015]. We define several variants of unfounded sets (one for each approximator of $T_\mathcal{P}$) and show that A-groundedness is directly characterised by means of A-unfounded sets. When taking for A Fitting's (partial) immediate consequence operator, our definition coincides with the notion of "3-unfounded set" [Bogaerts *et al.*, 2015].

Intuitively, an unfounded set is a set of atoms that might circularly support themselves, but have no support from outside. Stated differently, an unfounded set of a logic program $\mathcal{P}$ with respect to a partial interpretation $\mathcal{I}$ is a set U of atoms such that $\mathcal{P}$ provides no support for any atom in U if the atoms in U are assumed false. Each approximator A of the immediate consequence operator $T_\mathcal{P}$ defines its own notion of support: it maps a partial interpretation $\mathcal{I}$ to $A(\mathcal{I})$, where $A(\mathcal{I})_1$ are the atoms that are supported by $\mathcal{P}$, and $A(\mathcal{I})_2$ are the atoms that are possibly supported by $\mathcal{P}$, depending on the values of atoms unknown in $\mathcal{I}$. Thus, the atoms for which A provides no support are those not in $A(\mathcal{I})_2$ and the above intuitions are formalised directly as follows.

Definition 4.1 (A-Unfounded set)**.** Let $\mathcal{P}$ be a logic program, A an approximator of $T_\mathcal{P}$ and $\mathcal{I}$ a three-valued interpretation. A set $U \subseteq \Sigma$ is an *A-unfounded set* of $\mathcal{P}$ with respect to $\mathcal{I}$ if $A(\mathcal{I}[U : \mathbf{f}])_2 \cap U = \emptyset$.

When A is Fitting's immediate consequence operator, our definition coincides with the notion of 3-unfounded set defined by Bogaerts *et al.* [2015]. They showed that for partial interpretations $\mathcal{I}$ such that $\mathcal{I}[U : \mathbf{f}]$ is more precise than $\mathcal{I}$, $\Psi_\mathcal{P}$-unfounded sets are exactly unfounded sets as defined by Van Gelder *et al.* [1991] and that all interpretations in the well-founded model construction satisfy this condition.

We now show how groundedness relates to this generalised notion of unfounded sets. This proposition generalises Proposition 5.5 from Bogaerts *et al.* [2015].

Proposition 4.2. *Let $\mathcal{P}$ be a logic program, and A an approximator of the immediate consequence operator $T_\mathcal{P}$. A partial interpretation $\mathcal{I}$ is A-grounded if and only if all atoms that belong to an A-unfounded set U of P with respect to $\mathcal{I}$ are false in $\mathcal{I}$.*

Proof. First, suppose $\mathcal{I} = (I_1, I_2)$ is A-grounded and U is an A-unfounded set of $\mathcal{P}$ with respect to $\mathcal{I}$. Let $V = U^c$. Since U is an A-unfounded set, $A(\mathcal{I}[U : \mathbf{f}])_2 \cap U = \emptyset$. This means that $A(I_1 \wedge V, I_2 \wedge V)_2 \leq V$, hence the definition of A-groundedness yields $I_2 \leq V$: all elements of U are false in I. This must hold for every A-unfounded set U.

The reverse direction is analogous. Suppose every A-unfounded set is disjoint from I_2. Let V be such that $A(I_1 \wedge V, I_2 \wedge V)_2 \leq V$ and let $U = V^c$. Then again $\mathcal{I}[U : \mathbf{f}] = (I_1 \wedge V, I_2 \wedge V)$ hence U is an A-unfounded set of P with respect to $\mathcal{I}$. Thus, it must hold that $I_2 \cap U = \emptyset$, i.e., that $I_2 \leq V$. We conclude that $\mathcal{I}$ is indeed A-grounded. ☐

Following the analogy with the classical semantics of logic programs, we call a partial interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a $\Psi_\mathcal{P}$-grounded fixpoint of $T_\mathcal{P}$ a *partial grounded model* of $\mathcal{P}$. We briefly discuss complexity of partial grounded model semantics. First of all, deciding whether $\mathcal{P}$ has a partial grounded model is not an interesting task since the well-founded model is always grounded. We study other deterministic inference tasks, namely *credulous* and *sceptical* query answering (sometimes also called *possibility inference* and *certainty inference*, respectively) [Schlipf, 1995; Abiteboul *et al.*, 1990; Saccà, 1997] under the (partial) grounded semantics. The first of these tasks consists of deciding whether a symbol $p \in \Sigma$ holds in *some* grounded model, the second consists of deciding whether it holds in *all* grounded models.

Theorem 4.3. *Given a finite propositional logic program $\mathcal{P}$ over Σ and an atom $p \in \Sigma$, the following hold.*

1. *The problem of deciding whether p holds in* some *partial grounded model of $\mathcal{P}$ is Σ_2^P-complete.*
2. *The problem of deciding whether p holds in* all *partial grounded models of $\mathcal{P}$ is in P.*

The first of these points is proven analogous to the proof of Theorem 5.7 by Bogaerts *et al.* [2015]. Our proof is heavily inspired by theirs.

Proof. This first problem is in Σ_2^P; this follows from the fact that verifying whether a partial interpretation $\mathcal{I}$ is a partial grounded model can be done by calculating $\Psi_\mathcal{P}(\mathcal{I}[U : \mathbf{f}])$ for all candidate $\Psi_\mathcal{P}$-unfounded sets.

We now show Σ_2^P-hardness of the problem of existence of a (partial) grounded model of a program $\mathcal{P}$ in which p holds. Let φ be propositional formula in DNF over propositional symbols $x_1, \ldots, x_m, y_1, \ldots, y_n$. For an interpretation $I \subseteq \{x_1, \ldots, x_m\}$, we define φ_I as the formula obtained from φ by replacing all atoms $x_i \in I$ by $\mathbf{t}$ and all atoms $x_i \notin I$ by $\mathbf{f}$. Recall that the problem of deciding whether there exists an interpretation $I \subseteq \{x_1, \ldots, x_m\}$ such that φ_I is a tautology is Σ_2^P-hard. We now reduce this problem to our problem. For each x_i, we introduce a new variable x'_i with as intended meaning the negation of x_i. Let φ' be the formula obtained from φ by replacing all literals $\neg x_i$ by x'_i. Bogaerts *et al.* [2015] defined a program $\mathcal{P}(\varphi)$ consisting of the following clauses

1. $x_i \leftarrow \neg x'_i$ and $x'_i \leftarrow \neg x_i$ for each $i \in \{1, \ldots, m\}$,
2. $y_i \leftarrow \varphi'$ for each $i \in \{1, \ldots, n\}$,
3. $p \leftarrow \varphi'$,
4. $q \leftarrow \neg p \wedge \neg q$.

We need to show that there is an $I \subseteq \{x_1, \ldots, x_m\}$ such that φ_I is a tautology if and only if $P(\varphi)$ has a partial grounded model in which p holds.

Bogaerts *et al.* [2015] already showed that there is an $I \subseteq \{x_1, \ldots, x_m\}$ such that φ_I is a tautology if and only if $P(\varphi)$ has an exact grounded model. Hence, all we need to show is that if $P(\varphi)$ has a partial stable fixpoint, then there is such an I as well. We will prove a stronger claim, namely that every partial grounded model of $\mathcal{P}(\varphi)$ in which p holds can be extended to a two-valued grounded model in which p holds.

It is easy to see that in each partial fixpoint M of $\Psi_{\mathcal{P}(\varphi)}$ with $p^M = \mathbf{t}$ the following properties hold:

1. $y_1, \ldots, y_n$ are true in M (since their rules have the same body as p),
2. for each $i \in \{1, \ldots, m\}$, either both x_i and x'_i are unknown in M or one of them is true and the other is false in M.

Now, suppose M is a partial grounded model of $\mathcal{P}$. By the above observations, the only unknown atoms in M can be q, the x_i and the x'_i. Now, it is easy to see that $M[\{q\} : \mathbf{f}]$ is also a grounded model of $\mathcal{P}$. Similarly, if x_i and x'_i are unknown in M, then $M[\{x_i\} : \mathbf{t}, \{x'_i\} : \mathbf{f}]$ is a grounded model as well. Hence, our claim follows.

The second point follows from the fact that the well-founded model of $\mathcal{P}$ can be computed in polynomial time and that the well-founded model is the least precise partial grounded model (Corollary 3.7). □

5 Discussion and Future Work

Other classes of partial models In the context of logic programming and argumentation frameworks, many subclasses of partial stable models have been proposed.

- A partial stable model (X, Y) of $\mathcal{P}$ is called *M-stable* (for maximally stable) if it is $\leq_p$ maximal, i.e., if there is no more precise partial stable model [Saccà and Zaniolo, 1997]. These models coincide with the preferred extensions of Dung [1995] and the regular models of You and Yuan [1990].
- A partial stable model (X, Y) of $\mathcal{P}$ is called *L-stable* (for least undefined stable) if $Y \setminus X$ is $\subseteq$-minimal, i.e., if there exists no partial stable model (X', Y') of $\mathcal{P}$ such that $Y' \setminus X' \subsetneq Y \setminus X$ [Saccà and Zaniolo, 1997]. L-stable models only differ from exact stable models if a program has no two-valued stable models. This property is particularly useful in the context of debugging.

Recent work by Strass [2013] lifted these two notions to approximation fixpoint theory.[2] His definitions can be used immediately to define partial M-grounded and L-grounded fixpoints as well. Studying their properties, e.g., complexity, is a topic for future work.

Non-deterministic operators Pelov and Truszczynski [2004] characterised stable fixpoints of a disjunctive logic program using non-deterministic operators. Lots of work remains to be done in this area. Extending approximation fixpoint theory in general, and groundedness in particular to non-deterministic operators remains a topic for future work.

Applications In this paper, we only applied our theory to logic programming. However, as already indicated, our theory applies to all domains in which AFT is applied. Working out these applications is a topic for future work as well.

6 Conclusion

In this paper, we extended groundedness to points in the bilattice: for every approximating operator A, we defined a notion of A-groundedness. We showed that for exact lattice points, this coincides with groundedness for O. We related A-grounded fixpoints to other fixpoints in AFT: all A-stable fixpoints are A-grounded and the A-well-founded fixpoint is uniquely characterised as the least precise A-grounded fixpoint.

We applied our algebraical theory to logic programming, where A-groundedness is closely related to unfounded sets. Applying the theory to other research domains, such as default logic, autoepistemic logic and abstract dialectical frameworks is a topic for future work.

Acknowledgements We would like to thank the anonymous reviewers and Maurice Bruynooghe for their feedback on this work. This work was supported by the KU Leuven under project GOA 13/010 and by the Research Foundation - Flanders (FWO-Vlaanderen).

References

[Abiteboul *et al.*, 1990] Serge Abiteboul, Eric Simon, and Victor Vianu. Non-deterministic languages to express deterministic transformations. In *Proceedings of PODS*, pages 218–229, 1990.

[Antic *et al.*, 2013] Christian Antic, Thomas Eiter, and Michael Fink. Hex semantics via approximation fixpoint theory. In *Proceedings of LPNMR*, pages 102–115, 2013.

[Bi *et al.*, 2014] Yi Bi, Jia-Huai You, and Zhiyong Feng. A generalization of approximation fixpoint theory and application. In *Proceedings of RR*, pages 45–59, 2014.

[2]In order to define L-stable fixpoints, some additional assumptions about the lattice need to be made.

[Bogaerts *et al.*, 2014] Bart Bogaerts, Joost Vennekens, Marc Denecker, and Jan Van den Bussche. FO(C): A knowledge representation language of causality. *TPLP*, 14(4-5-Online-Supplement):60–69, 2014.

[Bogaerts *et al.*, 2015] Bart Bogaerts, Joost Vennekens, and Marc Denecker. Grounded fixpoints and their applications in knowledge representation. *AIJ*, 224:51–71, 2015.

[Brewka and Woltran, 2010] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proceedings of KR*, pages 102–111, 2010.

[Castro and Warren, 2001] Luís Fernando Castro and David Scott Warren. An environment for the exploration of non monotonic logic programs. In *Proceedings of WLPE*, 2001.

[Denecker and Vennekens, 2007] Marc Denecker and Joost Vennekens. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In *Proceedings of LPNMR*, pages 84–96, 2007.

[Denecker *et al.*, 2000] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In *Logic-Based Artificial Intelligence, Springer*, volume 597, pages 127–144, 2000.

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

[Denecker *et al.*, 2004] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation*, 192(1):84–121, July 2004.

[Denecker *et al.*, 2012] Marc Denecker, Maurice Bruynooghe, and Joost Vennekens. Approximation fixpoint theory and the semantics of logic and answers set programs. In *Correct Reasoning*, pages 178–194, 2012.

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

[Eiter *et al.*, 1997] Thomas Eiter, Nicola Leone, and Domenico Saccà. On the partial semantics for disjunctive deductive databases. *Ann. Math. AI*, 19(1-2):59–96, 1997.

[Fitting, 2002] Melvin Fitting. Fixpoint semantics for logic programming — A survey. *Theoretical Computer Science*, 278(1-2):25–51, 2002.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of ICLP/SLP*, pages 1070–1080, 1988.

[Kleene, 1938] S. C. Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938.

[Marek and Truszczyński, 1999] Victor Marek and Mirosław Truszczyński. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: A 25-Year Perspective*, pages 375–398. Springer-Verlag, 1999.

[Moore, 1985] Robert C. Moore. Semantical considerations on nonmonotonic logic. *AIJ*, 25(1):75–94, 1985.

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

[Pelov *et al.*, 2007] Nikolay Pelov, Marc Denecker, and Maurice Bruynooghe. Well-founded and stable semantics of logic programs with aggregates. *TPLP*, 7(3):301–353, 2007.

[Przymusinski, 1991] Teodor C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9(3/4):401–424, 1991.

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

[Saccà and Zaniolo, 1997] Domenico Saccà and Carlo Zaniolo. Deterministic and non-deterministic stable models. *J. Log. Comput.*, 7(5):555–579, 1997.

[Saccà, 1997] Domenico Saccà. The expressive powers of stable models for bound and unbound DATALOG queries. *J. Comput. Syst. Sci.*, 54(3):441–464, 1997.

[Schlipf, 1995] John S. Schlipf. Complexity and undecidability results for logic programming. *Annals of Mathematics and Artificial Intelligence*, 15(3-4):257–288, 1995.

[Seipel *et al.*, 1997] Dietmar Seipel, Jack Minker, and Carolina Ruiz. A characterization of the partial stable models for disjunctive databases. In *Proceedings of SLP*, pages 245–259, 1997.

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

[van Emden and Kowalski, 1976] Maarten H. van Emden and Robert A. Kowalski. The semantics of predicate logic as a programming language. *J. ACM*, 23(4):733–742, 1976.

[Van Gelder *et al.*, 1991] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *J. ACM*, 38(3):620–650, 1991.

[Vennekens *et al.*, 2006] Joost Vennekens, David Gilis, and Marc Denecker. Splitting an operator: Algebraic modularity results for logics with fixpoint semantics. *ACM Trans. Comput. Log.*, 7(4):765–797, 2006.

[Vennekens *et al.*, 2007] Joost Vennekens, Maarten Mariën, Johan Wittocx, and Marc Denecker. Predicate introduction for logics with a fixpoint semantics. Parts I and II. *Fundamenta Informaticae*, 79(1-2):187–227, September 2007.

[You and Yuan, 1990] Jia-Huai You and Li-Yan Yuan. Three-valued formalization of logic programming: Is it needed? In *Proceedings of PODS*, pages 172–182, 1990.