Safe Inductions: An Algebraic Study*

Bart Bogaerts and **Joost Vennekens** and **Marc Denecker**
KU Leuven, Department of Computer Science
Celestijnenlaan 200A, Leuven, Belgium

Abstract

In many knowledge representation formalisms, a constructive semantics is defined based on sequential applications of rules or of a semantic operator. These constructions often share the property that rule applications must be delayed until it is *safe* to do so: until it is known that the condition that triggers the rule will remain to hold. This intuition occurs for instance in the well-founded semantics of logic programs and in autoepistemic logic. In this paper, we formally define the safety criterion algebraically. We study properties of so-called *safe inductions* and apply our theory to logic programming and autoepistemic logic. For the latter, we show that safe inductions manage to capture the intended meaning of a class of theories on which all classical constructive semantics fail.

1 Introduction

In many fields of computational logic, natural forms of *induction* show up. Such an induction can be seen as a sequence of semantic structures obtained by iterative applications of rules or a semantic operator. For instance, in logic programming, it is natural to think of sequences of interpretations where at each stage a number of rules whose bodies are satisfied are triggered (i.e., their head is added to the current interpretation). For positive logic programs, all such sequences converge to the minimal model. For non-positive programs, this strategy may yield meaningless results. For instance, for the program $\mathcal{P} = \{a, \quad b \leftarrow \neg a\}$, one such sequence is $\emptyset, \{b\}, \{b, a\}$, the limit of which is not even a supported model of the logic program. Intuitively, what is wrong here is that the rule $b \leftarrow \neg a$ is applied before the value of a is established. For stratified programs, like $\mathcal{P}$, this problem has been resolved [Apt *et al.*, 1988]. For the general case, the well-founded semantics [Van Gelder *et al.*, 1991] offers a solution that uses three-valued interpretations instead of two-valued interpretations.

In recent work, the notions of *natural* and *safe* inductions for inductive definitions were introduced [Denecker and Vennekens, 2014; Denecker *et al.*, 2017]. It was argued that this kind of processes forms the essence of our understanding of inductive definitions.

In this paper, we lift those ideas to a more general setting: we provide a principled study of inductions in *approximation fixpoint theory (AFT)* (Denecker, Marek and Truszczyński (DMT) 2000), an algebraic theory that provides a unifying framework of semantics of nonmonotonic logics. We show convergence of safe inductions in general and study the relationship between safe inductions and various fixpoints defined in approximation fixpoint theory.

By presenting our theory in AFT, our results are broadly applicable. DMT [2000] originally developed AFT to unify semantics of logic programs, autoepistemic logic and default logic. Later, it was also used to define semantics of extensions of logic programs, such as HEX logic programs [Antic *et al.*, 2013] and an integration of logic programs with description logics [Liu *et al.*, 2016]. Strass [2013] showed that many semantics from Dung's argumentation frameworks (AFs) [Dung, 1995] and abstract dialectical frameworks (ADFs) [Brewka *et al.*, 2013] can be obtained by direct applications of AFT. Cruz-Filipe [2016] and Bogaerts and Cruz-Filipe [2017] showed that AFT has applications in database theory, for defining semantics of active integrity constraints [Flesca *et al.*, 2004].

The theory we present in this paper induces for each of the above logics notions of (safe) inductions and a *safe semantics*. Our complexity results are obtained for general operators and hence can also be transfered to various logics of interest. Throughout the paper, we give examples from logic programming; in Section 6, we apply our theory to autoepistemic logic. There, we show that safe inductions induce a constructive semantics that captures the intended semantics of a class of theories for which classical constructive semantics fail. This failure was recently exposed and solved using a notion of *set-inductions* which is based on sets of lattice elements instead of intervals (which are standard in AFT) [Bogaerts *et al.*, 2016]. We show that safe inductions provide an alternative solution to this problem. Our solution is more direct: in contrast to set-inductions or well-founded inductions [Denecker and Vennekens, 2007], safe inductions do not require any form of approximation; they are sequences in the original lattice. For logic programming, this means that they are sequences of interpretations such that some atoms are derived in each step. For AEL, this means that they are

*Bart Bogaerts is a postdoctoral fellow of the Research Foundation – Flanders (FWO).

sequences of possible world structures such that additional knowledge is derived in each step.

2 Preliminaries: Lattices and Operators

A *partially ordered set (poset)* $\langle L, \leq \rangle$ is a set L equipped with a partial order $\leq$, i.e., a reflexive, antisymmetric, transitive relation. We write $x < y$ for $x \leq y \wedge x \neq y$. We call $\langle L, \leq \rangle$ a *complete lattice* if every subset S of L has a least upper bound $\bigvee S$ and a greatest lower bound $\bigwedge S$. A complete lattice has a least element $\bot$ and a greatest element $\top$. We use notations $x \vee y = \bigvee(\{x, y\})$ and $x \wedge y = \bigwedge(\{x, y\})$.

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* of O 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 [Tarski, 1955], denoted $\mathrm{lfp}(O)$, which is also O's least prefixpoint and the limit of any *monotone induction*, i.e., of any increasing sequence $(x_i)_{i \geq 0}$ satisfying

- $x_0 = \bot$,
- $x_i \leq x_{i+1} \leq O(x_i)$, for successor ordinals $i + 1$,
- $x_\lambda = \bigvee(\{x_i \mid i < \lambda\})$, for limit ordinals λ.

Logic Programming 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 conjunction of literals 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 $T_\mathcal{P}$ [van Emden and Kowalski, 1976] that maps a structure I to the structure

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

This is an operator on the lattice $\langle 2^\Sigma, \subseteq \rangle$.

3 Safe Inductions

Let L be a lattice and O an operator on L, fixed throughout the rest of this paper.

Definition 3.1. We call $y \in L$ *derivable from* $x \in L$ if $x \leq y \leq x \vee O(x)$.

Definition 3.2. Let x be an element of L. An *O-induction in* x is a sequence $(x_i)_{i \leq \beta}$ such that

- $x_0 = x$,
- x_{i+1} is derivable from x_i for each $i < \beta$,
- $x_\lambda = \bigvee(\{x_i \mid i < \lambda\})$, for limit ordinals $\lambda \leq \beta$.

Intuitively, we view O as an operator that *constructs* certain lattice points. An O-induction is the associated construction process. Intuitively, if we are at a stage x_i, $O(x_i)$ represents what can be concluded from this given stage. Therefore, the next step x_{i+1} in the induction is at least x_i ($x_{i+1} \geq x_i$) and at most the combination of x_i and what can be concluded from it ($x_{i+1} \leq x_i \vee O(x_i)$). In the context of a powerset lattice (a lattice of the form $\langle 2^S, \subseteq \rangle$), this means that $x_{i+1} \subseteq x_i \cup O(x_i)$, i.e., x_{i+1} only contains elements that were already in x_i or such that O concludes them from x_i.

Definition 3.3. Let $\mathcal{N} = (x_i)_{i \leq \beta}$ and $\mathcal{N}' = (y_i)_{i \leq \alpha}$ be two O-inductions. We say that $\mathcal{N}'$ *extends* $\mathcal{N}$ if $\alpha \geq \beta$ and $x_i = y_i$ for all $i \leq \beta$. The extension is *strict* if $y_\alpha \neq x_\beta$.

Definition 3.4. An O-induction is *terminal* if there exists no O-induction that strictly extends it.

Proposition 3.5. *An O-induction $(x_i)_{i \leq \beta}$ is terminal if and only if x_β is a prefixpoint of O.*

Proposition 3.6. *If O is monotone, all monotone inductions are O-inductions in $\bot$ and vice versa.*

Corollary 3.7. *If O is monotone, all terminal O-inductions in $\bot$ converge to* $\mathrm{lfp}(O)$.

There is a high degree of non-determinism in O-inductions. For monotone operators O, despite this non-determinism, all O-inductions converge to the same point. As such, O-inductions provide (if O is monotone) a way to construct an intended lattice point ($\mathrm{lfp}\, O$). For non-monotone operators, the situation is quite different: O-inductions might not converge to a single point.

Example 3.8. Let $\mathcal{P}$ be the logic program

$$\{\ p, \qquad q \leftarrow \neg p\ \}$$

This is a simple, *stratified* logic program [Apt *et al.*, 1988; Przymusinski, 1988]. Its intended fixpoint (its so-called *perfect model*) is $\{p\}$. Let $T_\mathcal{P}$ denote its immediate consequence operator. The following are the three terminal strict $T_\mathcal{P}$-inductions in $\bot = \emptyset$.

$$\mathcal{N}_1 = (\emptyset, \{q\}, \{p, q\}) \quad \mathcal{N}_2 = (\emptyset, \{p, q\}) \quad \mathcal{N}_3 = (\emptyset, \{p\}) \ \blacktriangle$$

In Example 3.8 it can be seen that certain derivations in an O-induction happen prematurely. For instance, in $\mathcal{N}_1$ and $\mathcal{N}_2$, q is derived by the non-monotonic rule $q \leftarrow \neg p$. As soon as p is derived, this rule no longer applies: $q \notin T_\mathcal{P}(\{p, q\}) = \{p\}$. In this sequence, the rule was applied when it was not *safe* to do so. Below, we define a notion of safety to avoid such premature derivations, i.e., to only derive facts that remain derivable, regardless of what other derivations are made further on in the induction process.

Definition 3.9. Let x' be derivable from x. We say that x' is *safely derivable* from x if for each O-induction $(x_i)_{i \leq \beta}$ in x, it holds that $x' \leq x \vee O(x_\beta)$.

An O-induction $(x_i)_{i \leq \beta}$ is *safe* if x_{i+1} is safely derivable from x_i for each $i < \beta$.

In words, x' is safely derivable from x if no matter what other derivations we make (ending up in x_β), x' consists at most of what we have in x combined with what O concludes from x_β, i.e., $x' \leq x \vee O(x_\beta)$.

An induction is terminal if it cannot be extended into a strictly larger induction. We define a similar concept for safe inductions.

Definition 3.10. A safe O-induction $\mathcal{N}$ is *safe-terminal* if there exists no strict extension $\mathcal{N}'$ of $\mathcal{N}$ that is safe.

In Example 3.8, we showed that not all terminal O-inductions converge to the same lattice point. Luckily, the safety criterion warrants a better situation.

Theorem 3.11. *For each $x \in L$, all safe-terminal O-inductions in x converge to the same lattice point.*

In order to prove this theorem, we use the following result.

Lemma 3.12. *Let $\mathcal{N} = (x_i)_{i\leq\beta}$, $\mathcal{N}' = (y_i)_{i\leq\gamma}$ be two safe O-inductions with $x_0 = y_0$. For every $i \leq \beta, j \leq \gamma$ it holds that if $i{+}1 \leq \beta$ then $x_{i+1} \vee y_j$ is safely derivable from $x_i \vee y_j$ and if $j{+}1 \leq \gamma$ then $x_i \vee y_{j+1}$ is safely derivable from $x_i \vee y_j$.* “

Proof of Theorem 3.11. Let $\mathcal{N} = (x_i)_{i\leq\beta}$ and $\mathcal{N}' = (y_i)_{i\leq\gamma}$ be two safe-terminal O-inductions. Consider the sequence $(z_i)_{i\leq\beta+\gamma}$ where $z_i = x_i$ if $i \leq \beta$ and $z_{\beta+i} = x_\beta \vee y_i$ if $i \leq \gamma$. By Lemma 3.12, this sequence is a safe O-induction. Since $\mathcal{N}$ is safe-terminal, this sequence cannot be a strict extension of $\mathcal{N}$ and hence $x_\beta \vee y_\gamma = z_{\beta+\gamma} = x_\beta$, i.e., $y_\gamma \leq x_\beta$. A symmetric argument shows that $x_\beta \leq y_\gamma$, hence $x_\beta = y_\gamma$, as desired. □

Definition 3.13. The *safely defined point* by O, denoted $safe(O)$ is the limit of all safe-terminal O-inductions in $\bot$.

By Theorem 3.11, the safely defined point is well-defined. We now study some properties of the safely defined point.

Proposition 3.14. *For any operator O, $safe(O)$ is a postfixpoint of O, i.e., $safe(O) \leq O(safe(O))$.*

Example 3.15. Consider a lattice $\{\bot, \top\}$ with two elements and an operator O that maps $\bot$ to $\top$ and $\top$ to $\bot$. The safely defined point by O is $\bot$, since $\top$ is not safely derivable ($\top \not\leq \bot \vee O(\top)$). Here, the O-induction $\mathcal{N} = (\bot)$ is safe-terminal, but not terminal. ▲

Definition 3.16. We call an operator O *complete* if the safely defined point by O is a fixpoint of O, i.e., if $O(safe(O)) = safe(O)$.

We will be mostly interested in complete operators O, as they uniquely determine a fixpoint of interest of O.

Proposition 3.17. *An operator O is complete if and only if every safe-terminal O-induction in $\bot$ is terminal.*

Proposition 3.18. *If O is a monotone operator, then O is complete and $safe(O) = \mathrm{lfp}(O)$.*

Theorem 3.11 shows that safe O-inductions, despite their non-determinism, uniquely determine a lattice point of interest. Furthermore, if O is monotone, this point is the least fixpoint of O. The question now arises: what if O is non-monotone? How does the safely defined point by O relate to other points of interest? In particular, how does it relate to fixpoints defined in approximation fixpoint theory? We study this in Section 5. First, we study complexity.

Complexity The *height* of a finite lattice L is the length n of the longest sequence $\bot = x_0 < x_1 < \cdots < \top = x_n$ in L. We call $y \in L$ a *direct successor* of $x \in L$ if $x < y$ and there is no z such that $x < z < y$. The *branching width* of a finite L is the maximum over $x \in L$ of the number of direct successors of x. All complexity results presented below are in terms of the sum of the branching width and the height of the input lattice. This means that we use the sum of the branching width and the height as the measure of our input.

In this section, we assume that a class $\mathcal{C} = \{\langle L, O\rangle\}$ of pairs of a finite lattice L and an operator $O : L \to L$ is given. Let $F_\mathcal{C}$ denote the function problem: given one of the $\langle L, O\rangle$ in $\mathcal{C}$ and $p, p' \in L$, compute **(1)** $O(p)$, **(2)** $p \vee p'$, and **(3)** $\{x \mid x \text{ is a direct successor of } p\}$. We assume that $F_\mathcal{C}$ can be solved in polynomial time.

The kind of setting used here is not so unusual: it is an algebraic variant of data complexity. For instance, in logic programming, each non-ground program $\mathcal{P}$ determines a class of lattices and associated operators (immediate consequence operators of the groundings of $\mathcal{P}$ with respect to a given domain). The height and branching width of the lattice are then polynomial in terms of the domain size. In this setting, the problem $F_{\mathcal{C}_\mathcal{P}}$ is indeed polynomially solvable.

Theorem 3.19. *Let $\mathcal{C}$ be a class as above. The decision problem* **given** $L_i \in \mathcal{C}, x, y \in L_i$**, is** y **safely derivable from** x **by** O_i**?** *is in co-NP.*

Sketch of the proof. To prove this, we build a program that nondeterministically traverses an O-induction from x. To determine that y is not safely derivable from x, it suffices to find one run of the algorithm such that $y \not\leq x \vee O(s)$ with s a state in in the O-induction. Such algorithm runs in polynomial time in the height of the lattice. □

Theorem 3.20. *Let $\mathcal{C}$ be a class as above. The decision problem* **given** $\langle L, O\rangle \in \mathcal{C}, s \in L$**, is** $safe(O) \geq s$**?** *is in (Δ_2^P). For some classes $\mathcal{C}$, this problem is co-NP-hard.*

Sketch of the proof. Containment follows from Theorem 3.19. Hardness follows from Theorem 6.10 of Denecker *et al.* [2017]. □

4 Preliminaries: AFT

Given a lattice L, approximation fixpoint theory makes use of the lattice 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 $x \leq y$. We use L^c to denote the set of consistent elements. Elements $(x, x) \in L^c$ are called *exact*. We sometimes use the tuple (x, y) and the interval $[x, y]$ interchangeably. The *precision ordering* on L^2 is defined as $(x, y) \leq_p (u, v)$ if $x \leq u$ and $v \leq y$. In case (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 $\langle L^2, \leq_p\rangle$ is also a complete lattice.

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 $O(x) \in A(x, x)$ for all $x \in L$. Approximators 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 (defined below) 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 $y = \text{lfp}(A(x,\cdot)_2)$, 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. This is equivalent to the condition that $x = \text{lfp}(A(\cdot,x)_1)$. A-stable fixpoints are minimal fixpoints of O. The A-Kripke-Kleene fixpoint of O can be constructed by iterative applications of A, starting from $(\bot,\top)$. For the A-well-founded fixpoint, a similar constructive characterization has been worked out.

Definition 4.1. 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 4.2. 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 $i < \beta$;
- $(x_\lambda,y_\lambda) = \bigvee_{\leq_p}\{(x_i,y_i) | i < \lambda\}$ for limit ordinals $\lambda \leq \beta$.

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

Denecker and Vennekens [2007] showed that all terminal A-inductions converge to the A-well-founded fixpoint of O.

Logic Programming For logic programming, DMT showed that Fitting's immediate consequence operator $\Psi_\mathcal{P}$ [Fitting, 2002] is an approximator of $T_\mathcal{P}$, 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 the stable models of $\mathcal{P}$ [Gelfond and Lifschitz, 1988].

5 Safe Inductions and AFT

In this section, we study how (safe) O-inductions relate to the fixpoints studied in AFT.

Theorem 5.1. *Let O be an operator and A an approximator of O. The A-well-founded fixpoint approximates the safely defined point by O.*

The proof makes use of the following proposition.

Proposition 5.2. *Let O be an operator and A an approximator of O. Let $(x_i,y_i)_{i\leq\beta}$ be an A-well-founded induction. The following claims hold:*

(1) *$(x_i)_{i\leq\beta}$ is a safe O-induction, and*

(2) *for each $i \leq \beta$ and each O-induction $\mathcal{N} = (z_j)_{j\leq\alpha}$ with $z_0 = x_i$, it holds that $z_\alpha \leq y_\beta$*

Proof of Theorem 5.1. Let z denote the safely defined point of O and let (x_β,y_β) denote the A-well-founded fixpoint of O. For any terminal A-well-founded induction $(x_i,y_i)_{i\leq\beta}$, it holds that $x_\beta \leq z$ by the first point of Proposition 5.2. Furthermore, by the second point of Proposition 5.2 it holds that any O-induction stays under y_β; hence $z \leq y_\beta$. □

Theorem 5.1 has several consequences.

Corollary 5.3. *If the A-well-founded fixpoint of O is exact, i.e., equal to (x,x) for some $x \in L$, then O is complete and $safe(O) = x$.*

Corollary 5.4. *Let O be an operator and A an approximator of O. The A-Kripke-Kleene fixpoint of O approximates the safely defined point by O.*

Corollary 5.5. *If the A-Kripke-Kleene fixpoint of O is exact, i.e., equal to (x,x) for some $x \in L$, then O is complete and $safe(O) = x$.*

Safe O-inductions identify a unique lattice point of interest. Since an operator can have multiple stable fixpoints, we cannot expect a strong link between the safely defined point and stable fixpoints. However, we do find the following relation between stable fixpoints and O-inductions.

Theorem 5.6. *Let A be an approximator of O. If x is an A-stable fixpoint of O, then x is the limit of a terminal O-induction.*

Sketch of the proof. If x is an A-stable fixpoint of O, then $x = \text{lfp}(A(\cdot,x)_1)$. The clue to proving this proposition is to show that monotone inductions of $A(\cdot,x)_1$ are O-inductions. The result then easily follows. □

Example 5.7. Consider the logic program

$$\mathcal{P} = \{\ p \leftarrow \neg q, \qquad q \leftarrow \neg p\ \}$$

It holds that $\{p\}$ is a stable model of $\mathcal{P}$ (a $\Psi_\mathcal{P}$-stable fixpoint of $T_\mathcal{P}$). Also, $\{p\}$ is the limit of the $T_\mathcal{P}$-induction $(\emptyset,\{p\})$. This induction is not safe since $(\emptyset,\{q\})$ is also a $T_\mathcal{P}$-induction and $\{p\} \not\leq T_\mathcal{P}(\{q\}) \vee \emptyset = \{q\}$. ▲

The limit of a terminal O induction is not always a stable fixpoint of O (for some approximator A), as we show below.

Example 5.8. Consider the logic program

$$\mathcal{P} = \left\{ \begin{array}{ll} p \leftarrow p, & p \leftarrow q, \\ q \leftarrow \neg p, & q \leftarrow q \end{array} \right\}$$

In this case $(\emptyset,\{q\},\{q,p\})$ is the unique terminal $T_\mathcal{P}$-induction. It can be verified that this is a safe induction and that $T_\mathcal{P}$ is complete. The safely defined point is a non-minimal fixpoint of $T_\mathcal{P}$, hence it is also non-grounded (see [Bogaerts *et al.*, 2015]) and not an A-stable fixpoint for any approximator A of $T_\mathcal{P}$. In the well-founded model of $\mathcal{P}$, all atoms are unknown. ▲

6 Safe Inductions and Autoepistemic Logic

Recently Bogaerts *et al.* [2016] exposed a problem in several semantics of autoepistemic logic (AEL). They showed that for very simple, stratified theories, the well-founded and other semantics fail to identify the intended model. They solved this problem by defining, algebraically, a new constructive semantics that is based on a refined notion of approximations of a lattice point (more refined than intervals, i.e., elements of L^2). In this section, we show that safe inductions provide a direct solution to the aforementioned problem without the need for any approximation. First, we recall some background on AEL.

6.1 AFT and Autoepistemic Logic

AEL is a non-monotonic logic for modeling the beliefs or knowledge of a rational agent with perfect introspection capabilities [Moore, 1985].

Let $\mathcal{L}$ be the language of propositional logic based on a set of atoms Σ. Extending this language with a modal operator K, which is read "I (the agent) know"[1], yields a language $\mathcal{L}_K$ of modal propositional logic. An *autoepistemic theory* is a set of formulas in $\mathcal{L}_K$. A crucial assumption about such theories that distinguishes this logic from the standard modal logic S5 is that all of the agent's knowledge is encoded in the theory: it either belongs to the theory, or can be derived from it. Levesque [1990] called this the "all I know assumption".

A *modal formula* is a formula of the form $K\psi$; an *objective formula* is a formula without modal subformulas. If φ is a formula, $At(\varphi)$ denotes the set of all atoms that occur in φ and $At_O(\varphi)$ the set of all atoms that occur objectively in φ, i.e., outside of the scope of an operator K.

An *interpretation* is a subset of Σ. A *possible world structure* is a set of interpretations. A possible world structure can be seen as a Kripke structure in which the accessibility relation is total. The set of all possible world structures is denoted $\mathcal{W}_\Sigma$; it forms a lattice with the knowledge order $\leq_k$ such that $Q \leq_k Q'$ iff $Q \supseteq Q'$. A possible world structure Q is a mathematical object to represent all situations that are possible according to the agent: interpretations $q \in Q$ represent possible states of affairs, i.e., states of affairs consistent with the agent's knowledge, and interpretations $q \notin Q$ represent impossible states of affairs, i.e., states of affairs that violate the agent's knowledge.

If φ is a formula in $\mathcal{L}_K$, Q is a possible world structure and I is an interpretation, satisfaction of φ with respect to Q and I (denoted $Q, I \models \varphi$) is defined as in the modal logic S5 by the standard recursive rules of propositional satisfaction augmented with one additional rule:

$$Q, I \models K\varphi \text{ if } Q, I' \models \varphi \text{ for every } I' \in Q.$$

In this formula, Q represents the belief of the agent and I represents the actual state of the world. Modal formulas are evaluated with respect to the agent's belief, while objective formulas are evaluated with respect to the state of the actual world. We furthermore define $Q \models K\varphi$ (φ is known in Q) if $Q, I \models \varphi$ for every $I \in Q$. Moore [1985] associated with every theory $\mathcal{T}$ an operator $D_\mathcal{T}$ on $\mathcal{W}_\Sigma$ as follows:

$$D_\mathcal{T}(Q) = \{I \in \mathcal{W}_\Sigma \mid Q, I \models \mathcal{T}\}.$$

The intuition behind this operator is that $D_\mathcal{T}(Q)$ is a revision of Q consisting of all worlds that are consistent with the agent's current beliefs (Q) and the constraints in $\mathcal{T}$.

DMT [2003] defined approximators for $D_\mathcal{T}$ and showed that AFT induces all main and some new semantics for AFT.

Monotonically Stratified AEL Theories Following Vennekens *et al.* [2006], we call an autoepistemic theory $\mathcal{T}$ *stratifiable*[2] w.r.t. a partition $(\Sigma_i)_{0\leq i\leq n}$ of its alphabet if there exists a partition $(\mathcal{T}_i)_{0\leq i\leq n}$ of $\mathcal{T}$ such that for each i, $At_O(\mathcal{T}_i) \subseteq \Sigma_i$ and $At(\mathcal{T}_i) \subseteq \bigcup_{0\leq j\leq i} \Sigma_j$. This notion of stratification significantly extends the notion from Marek and Truszczyński [1991]. A stratification is *modally separated* if for every modal subformula $K\psi$ of $\mathcal{T}_i$, either $At(\psi) \subseteq \Sigma_i$ or $At(\psi) \subseteq \bigcup_{0\leq j<i} \Sigma_j$.

Let Σ_1 and Σ_2 be two disjoint vocabularies. If Q_1 and Q_2 are possible world structures over Σ_1 and Σ_2 respectively, then the extension of Q_1 by Q_2 is the possible world structure over $\Sigma_1 \cup \Sigma_2$ defined as $Q_1 \oplus Q_2 \stackrel{\text{def}}{=} \{I_1 \cup I_2 \mid I_1 \in Q_1 \wedge I_2 \in Q_2\}$. If Q is a possible world structure over $\Sigma_1 \cup \Sigma_2$, the restriction of Q to Σ_1 is $Q|_{\Sigma_1} \stackrel{\text{def}}{=} \{I \cap \Sigma_1 \mid I \in Q\}$.

DMT [2011] have made strong arguments in favor of a constructive semantics for AEL. Bogaerts *et al.* [2016], however, showed that the two constructive semantics induced by AFT (well-founded and Kripke-Kleene semantics) are too weak for AEL. They gave the following example.

Example 6.1. Consider the autoepistemic theory

$$\mathcal{T} = \{q \Leftrightarrow \neg Kp, r \Leftrightarrow \neg Kq\}.$$

The informal reading of this theory is as follows: I (an introspective autoepistemic agent) only know the following: q holds iff I do not know p and r holds iff I do not know q.

Since p does not occur objectively in $\mathcal{T}$, an agent who only knows $\mathcal{T}$ does not have any information about p. Thus, in the intended model, it knows neither p nor $\neg p$, i.e., $\neg Kp$ and $\neg K\neg p$ must hold in the intended model. The first sentence then entails q, hence Kq must hold. Now, the last sentence implies $\neg r$; the intended model is thus $\{\{p, q\}, \{q\}\}$, the unique possible world structure in which $\neg Kp, \neg K\neg p, Kq$, and $K\neg r$ hold. ▲

Bogaerts *et al.* [2016] showed that the well-founded semantics (for any approximator) fails to identify the intended model in the above example. They generalized this example to the class of *monotonically stratified* theories and defined a notion of *perfect model* for them.

Definition 6.2. We say that $\mathcal{T}$ is *monotonically stratified* with respect to a partition $(\Sigma_i)_{0\leq i\leq n}$ of its alphabet if there is a modally separated stratification $(\mathcal{T}_i)_{0\leq i\leq n}$ of $\mathcal{T}$ such that all subformulas $K\psi$ of $\mathcal{T}_i$ with $At(\psi) \subseteq \Sigma_i$ occur negatively (in the scope of an odd number of negations) in $\mathcal{T}_i$.

The construction of the perfect model of an autoepistemic theory is as follows. In a monotonically stratified theory, each theory $\mathcal{T}_i$ defines knowledge of the symbols in Σ_i in terms of knowledge of symbols in lower strata (Σ_j with $j < i$). The last condition guarantees that for a fixed interpretation of the knowledge of lower strata, $D_{\mathcal{T}_i}$ is a monotone operator and hence its intended fixpoint is clear. The perfect model of $\mathcal{T}$ is then constructed by iterated monotone inductions, each of them computing the knowledge of symbols in Σ_i based on the knowledge of symbols in lower strata. In the example above, first ignorance of p is established; next, knowledge of q is established and in the final stage, knowledge of $\neg r$ is concluded. This construction was formalized as follows.

Proposition 6.3 (Proposition 3.3 from Bogaerts *et al.* [2016])**.** *Let $(\mathcal{T}_i)_{0\leq i\leq n}$ be a monotonic stratification of $\mathcal{T}$ w.r.t.*

[1] Or, following DMT [2011] "My knowledge entails".

[2] As mentioned in the introduction, we restrict to finite stratifications here.

$(\Sigma_i)_{0\leq i\leq n}$. *For some i, let Q_{i-1} be a possible world structure over $\bigcup_{j<i}\Sigma_j$. The operator $D_i : \mathcal{W}_{\Sigma_i} \to \mathcal{W}_{\Sigma_i} : Q \mapsto D_{\mathcal{T}_i}(Q \oplus Q_{i-1})|_{\Sigma_i}$ is monotone.*

Definition 6.4. Let $\mathcal{T}$ be a monotonically stratified autoepistemic theory and $(\mathcal{T}_i)_{0\leq i\leq n}$ a monotonic stratification of $\mathcal{T}$. The *perfect model* of $\mathcal{T}$ (denoted $pm(\mathcal{T})$) is defined by induction on n.

- If $n = 0$, then $D_{\mathcal{T}}$ is monotone and the perfect model of $\mathcal{T}$ is the least fixpoint of $D_{\mathcal{T}}$.
- Otherwise, let Q_{n-1} denote $pm(\bigcup_{j<n}\mathcal{T}_j)$ and let D_n be as in Proposition 6.3; in this case we define $pm(\mathcal{T})$ as $\mathrm{lfp}(D_n) \oplus Q_{n-1}$.

In general, the construction of the perfect model may not always work as expected. Bogaerts *et al.* [2016] defined a criterion that guarantees that this construction behaves nicely, called *weak permaconsistency*.

Definition 6.5. An autoepistemic theory $\mathcal{T}$ is called *weakly permaconsistent* if for every possible world structure Q, there is at least one I such that $Q, I \models \mathcal{T}$.

This resulted in a "sanity criterion" for semantics of autoepistemic logic as follows.

Definition 6.6. We say that a semantics for autoepistemic logic *respects stratification* if all weakly permaconsistent monotonically stratified theories have exactly one model, namely their perfect model.

6.2 AEL and Safe Inductions

Here, we show that the safely defined point of $D_{\mathcal{T}}$ manages to identify the fixpoint of interest for Example 6.1 and that this result generalizes: the safely defined semantics (defined formally below) respects stratification. This result shows that safe inductions can identify the perfect model, *without prior information on the stratification* and *without the need for any form of approximation*. Even stronger, the perfect model construction *is* a terminal safe induction.

Definition 6.7. The *safely defined semantics* is given by $Q \models_{sd} \mathcal{T}$ if $Q = safe(D_{\mathcal{T}})$ and $D_{\mathcal{T}}$ is complete.

The condition that $D_{\mathcal{T}}$ is complete has as effect here that the safely defined model of $\mathcal{T}$ must be a fixpoint of $D_{\mathcal{T}}$. In other words, the knowledge of the agent must be such that it can no longer be revised by the revision operator.

Example 6.8 (Example 6.1 continued)**.** A first observation is that there are no possible world structures Q such that $D_{\mathcal{T}}(Q) \models Kp$ or $D_{\mathcal{T}}(Q) \models K\neg p$. Hence, if $\mathcal{N} = (Q_i)_{i\leq\beta}$ is a $D_{\mathcal{T}}$-induction in $\bot = 2^{\{p,q,r\}}$, it also has the property that $Q_i \not\models Kp$ and $Q_i \not\models K\neg p$ for each i. For each Q_i, it then holds that $D_{\mathcal{T}}(Q_i) \models Kq$. From this it follows that $Q_q := \{\{p,q\},\{q\},\{p,q,r\},\{q,r\}\}$, the $\leq_k$-least possible world structure in which Kq holds, is safely derivable from $\bot$. Now, for every possible world structure $Q \geq_k Q_q$, it holds that $D_{\mathcal{T}}(Q) \models K\neg r$. Thus, this also holds for all possible world structures in a $D_{\mathcal{T}}$-induction from Q_q. Hence, it follows that $\{\{p,q\},\{q\}\}$ is safely derivable from Q_q. Since this is a fixpoint of $D_{\mathcal{T}}$, the safe $D_{\mathcal{T}}$-induction

$$(\bot, Q_q, \{\{p,q\},\{q\}\})$$

is terminal and hence also safe-terminal. Thus, the perfect model of $\mathcal{T}$ is indeed the safely defined point by $D_{\mathcal{T}}$. ▲

We now give a sketch of the proof that the above example is not a coincidence, i.e., that it generalizes to the class of monotonically stratified theories.

Theorem 6.9. *The safely defined semantics respects stratification. That is: for each monotonically stratified theory $\mathcal{T}$: if $\mathcal{T}$ is weakly permaconsistent, then $D_{\mathcal{T}}$ is complete and $safe(D_{\mathcal{T}})$ is the perfect model of $\mathcal{T}$.*

The proof of this theorem makes use of the following two results.

Lemma 6.10. *Suppose $(\mathcal{T}_i)_{0\leq i\leq n}$ is a monotone stratification of $\mathcal{T}$ w.r.t. $(\Sigma_i)_{0\leq i\leq n}$. Let Σ'_i denote $\bigcup_{j\leq i}\Sigma_j$ for each i. For every possible world structure Q it holds that*

$$D_{\mathcal{T}}(Q) = \bigoplus_{0\leq i\leq n} D_{\mathcal{T}_i}(Q|_{\Sigma'_i})|_{\Sigma_i}.$$

Lemma 6.11. *Suppose $\mathcal{T}$ is monotonically stratified w.r.t. $(\Sigma_i)_{0\leq i\leq n}$. Furthermore suppose $\mathcal{T}$ is weakly permaconsistent. Let Σ'_i denote $\bigcup_{j\leq i}\Sigma_i$ for each i. If Q_1 and Q_2 are two possible world structures such that $Q_1|_{\Sigma'_i} = Q_2|_{\Sigma'_i}$, then also $D_{\mathcal{T}}(Q_1)|_{\Sigma'_i} = D_{\mathcal{T}}(Q_2)|_{\Sigma'_i}$.*

Lemma 6.10 shows how $D_{\mathcal{T}}$ is composed from the various $D_{\mathcal{T}_i}$. Lemma 6.11 states that if two possible world structures agree on the lower strata, then so does their image under $D_{\mathcal{T}}$ for any weakly permaconsistent theory $\mathcal{T}$. In other words: the knowledge of symbols in a given stratum in $D_{\mathcal{T}}(Q)$ only depends on the knowledge of symbols of smaller (or equal) strata in Q.

Sketch of the proof of Theorem 6.9. The central idea in this proof is to turn the construction of the perfect model into a safe $D_{\mathcal{T}}$-induction. To show that it is a $D_{\mathcal{T}}$-induction and to show that it is safe, we repeatedly exploit the fact that if Q agrees with $pm(\mathcal{T})$ on all strata below i, so does $D_{\mathcal{T}}(Q)$.

Completeness follows from the fact that we find a safe $D_{\mathcal{T}}$ induction whose limit is the perfect model of $\mathcal{T}$. Since this model is a fixpoint of $D_{\mathcal{T}}$, this induction is terminal and $D_{\mathcal{T}}$ is indeed complete. □

7 Conclusion

In this paper, we presented the notions of O-inductions and safe O-inductions for a lattice operator O. We studied how they relate to various fixpoints of O studied in AFT. We studied the semantics induced by these inductions in the context of autoepistemic logic, where we find that the safely defined point has interesting properties for a class of operators. It is a topic of future work to study the semantics induced by safe inductions for other application domains of AFT, such as abstract argumentation [Dung, 1995] and active integrity constraints [Flesca *et al.*, 2004; Cruz-Filipe, 2016], where we conjecture that safe inductions will prove helpful to tackle the problems with the well-founded semantics such as Example 18 of Cruz-Filipe [2016].

References

[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.

[Apt *et al.*, 1988] Krzysztof R. Apt, Howard A. Blair, and Adrian Walker. Towards a theory of declarative knowledge. In Minker [1988], pages 89–148.

[Bogaerts and Cruz-Filipe, 2017] Bart Bogaerts and Luís Cruz-Filipe. Semantics for active integrity constraints using approximation fixpoint theory. In *Proceedings of IJCAI*, 2017. (to appear).

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

[Bogaerts *et al.*, 2016] Bart Bogaerts, Joost Vennekens, and Marc Denecker. On well-founded set-inductions and locally monotone operators. *ACM Trans. Comput. Logic*, 17(4):27:1–27:32, September 2016.

[Brewka *et al.*, 2013] Gerhard Brewka, Hannes Strass, Stefan Ellmauthaler, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proceedings of IJCAI*, 2013.

[Cruz-Filipe, 2016] Luís Cruz-Filipe. Grounded fixpoints and active integrity constraints. In *Technical communications of ICLP*, pages 11.1–11.14, 2016.

[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 and Vennekens, 2014] Marc Denecker and Joost Vennekens. The well-founded semantics is the principle of inductive definition, revisited. In *Proceedings of KR*, pages 22–31, 2014.

[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.*, 2011] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Reiter's default logic is a logic of autoepistemic reasoning and a good one, too. In *Nonmonotonic Reasoning – Essays Celebrating Its 30th Anniversary*, pages 111–144. College Publications, 2011.

[Denecker *et al.*, 2017] Marc Denecker, Joost Vennekens, and Bart Bogaerts. A logical study of some common principles of inductive definition and its implications for knowledge representation. *CoRR*, abs/1702.04551, 2017.

[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.

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

[Flesca *et al.*, 2004] Sergio Flesca, Sergio Greco, and Ester Zumpano. Active integrity constraints. In *Proceedings of SIGPLAN*, pages 98–107, 2004.

[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.

[Levesque, 1990] Hector J. Levesque. All I know: A study in autoepistemic logic. *AIJ*, 42(2-3):263–309, 1990.

[Liu *et al.*, 2016] Fangfang Liu, Yi Bi, Md. Solimul Chowdhury, Jia-Huai You, and Zhiyong Feng. Flexible approximators for approximating fixpoint theory. In *Proceedings of Canadian AI*, pages 224–236, 2016.

[Marek and Truszczyński, 1991] Victor Marek and Mirosław Truszczyński. Autoepistemic logic. *J. ACM*, 38(3):588–619, 1991.

[Minker, 1988] Jack Minker, editor. *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, 1988.

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

[Przymusinski, 1988] Teodor C. Przymusinski. On the declarative semantics of deductive databases and logic programs. In Minker [1988], pages 193–216.

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

[Tarski, 1955] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 1955.

[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.