Safe Inductions and Their Applications in Knowledge Representation[☆]

Bart Bogaerts[a,*], Joost Vennekens[a,b], Marc Denecker[a]

[a] *Department of Computer Science, KU Leuven, 3001 Heverlee, Belgium*
[b] *Department of Computer Science, KU Leuven, Campus De Nayer, 2860 Sint-Katelijne-Waver, 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 continue 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.

Keywords: approximation fixpoint theory, lattice operator, inductive definitions, induction process, construction, well-founded semantics, groundedness, logic programming, autoepistemic logic, abstract argumentation

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

$$\mathcal{P} = \left\{ \begin{array}{l} a \\ b \leftarrow \neg a \end{array} \right\},$$

one such sequence is

$$\mathcal{N}_1 = \emptyset, \{b\}, \{b, a\},$$

the limit of which is not even a supported model of the logic program. On the other hand, the sequence

$$\mathcal{N}_2 = \emptyset, \{a\}$$

is another such sequence that *does* end in the intended model of $\mathcal{P}$, namely its perfect model. Intuitively, what is wrong with $\mathcal{N}_1$ is that the rule $b \leftarrow \neg a$ is applied too soon, before the value of a is established. For

[☆] A short version of this paper is accepted for publication in the proceedings of the IJCAI'17 conference (Bogaerts et al., 2017). This paper extends the previous work with more theoretical results, examples, proofs of all propositions and applications of the work to argumentation frameworks.

[*] Corresponding author

Email addresses: bart.bogaerts@cs.kuleuven.be (Bart Bogaerts), joost.vennekens@cs.kuleuven.be (Joost Vennekens), marc.denecker@cs.kuleuven.be (Marc Denecker)

stratified programs, like $\mathcal{P}$, this problem has been resolved, e.g., by 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 process forms the essence of our understanding of inductive definitions.

In this paper, we lift those ideas of safe and natural inductions to a more general setting: we provide a principled study of such inductions in the context of *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 this general setting and study the relationship between (algebraic) 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 (van Emden and Kowalski, 1976), autoepistemic logic (Moore, 1985) and default logic (Reiter, 1980). 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 for Dung's argumentation frameworks (AFs) (Dung, 1995) and abstract dialectical frameworks (ADFs) (Brewka et al., 2013) can be obtained by direct application of AFT. Bogaerts and Cruz-Filipe (2018) 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 transferred to various logics of interest. Throughout the paper, we give examples from logic programming.

In Section 7, 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 sequences of possible world structures such that additional knowledge is derived in each step.

In Section 8, we apply our theory to Dung's argumentation frameworks (Dung, 1995), where we show the surprising result that two different operators that exist for a given argumentation framework have *the same* safely defined point. Furthermore, this point corresponds to an existing semantics: it is the so-called *grounded extension*.

The rest of this paper is structured as follows. In Section 2, we give preliminaries regarding lattices and operators. In Section 3, we define (safe) inductions and provide some basic results. We continue by studying complexity of some inference problems related to safe inductions in Section 4. In Section 5, we recall the basics of AFT; we use this in Section 6 tos study how (safe) inductions relate to various fixpoints studied in AFT. Afterwards, in Sections 7 and 8, we apply our general theory to autoepistemic logic and argumentation frameworks respectively. We conclude in Section 9.

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, anti-symmetric, transitive relation. We write $x < y$ for $x \leq y \wedge x \neq y$. If S is a subset of L, then x is an *upper bound*, respectively a *lower bound* of S if for every $s \in S$, it holds that $s \leq x$, respectively $x \leq s$. An element x is a *least upper bound*, respectively *greatest lower bound* of S if it is an upper bound that is smaller than every other upper bound, respectively a lower bound that is greater than every other lower bound. If S

has a least upper bound, respectively a greatest lower bound, we denote it $lub(S)$, respectively $glb(S)$. As is custom, we sometimes call a greatest lower bound a *meet*, and a least upper bound a *join* and use the related notations $\bigwedge S = glb(S)$, $x \wedge y = glb(\{x, y\})$, $\bigvee S = lub(S)$ and $x \vee y = lub(\{x, y\})$. We call $\langle L, \leq \rangle$ a *complete lattice* if every subset S of L has a least upper bound and a greatest lower bound. A complete lattice has a least element $\bot$ and a greatest element $\top$.

An operator $O : L \to L$ is *monotone* if $x \leq y$ implies that $O(x) \leq O(y)$ and *anti-monotone* if $x \leq y$ implies that $O(y) \leq O(x)$. 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 terminal *monotone induction* of O, defined below.

Definition 2.1. A *monotone induction* of a lattice operator $O : L \to L$ is an increasing sequence (for some ordinal β) $(x_i)_{i \leq \beta}$ of elements $x_i \in L$ satisfying

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

A monotone induction is *terminal* if $O(x_\beta) = x_\beta$.

Logic Programming. Let Σ be an alphabet, i.e., a collection of symbols which are called *atoms*. A *literal* is an atom p or the negation $\neg q$ of an atom q. The former are called *positive* literals; the latter are called *negative* literals. 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 $\langle 2^\Sigma, \subseteq \rangle$. We call a logic program $\mathcal{P}$ *positive* if for each rule $r \in \mathcal{P}$, $body(r)$ consists of only positive literals. If $\mathcal{P}$ is positive, then $T_\mathcal{P}$ is monotone.

3. Safe Inductions

In this section, we define the central concept of this paper, namely the notion of a *safe induction* and study its basic properties. 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 $i < \beta$,
- $x_\lambda = lub(\{x_i \mid i < \lambda\})$, for limit ordinals $\lambda \leq \beta$.

We call x_β the *limit* of $(x_i)_{i \leq \beta}$.

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.*

Proof. Let $\mathcal{N}$ denote $(x_i)_{i \leq \beta}$.

If x_β is a prefixpoint of O, then $O(x_\beta) \leq x_\beta$, hence, in each extension $(x_i)_{i \leq \alpha}$ with $\alpha > \beta$ of $\mathcal{N}$, it must hold that

$$x_\beta \leq x_{\beta+1} \leq x_\beta \vee O(x_\beta) = x_\beta,$$

hence $x_{\beta+1} = x_\beta$ and by induction also $x_\alpha = x_\beta$.

On the other hand, if x_β is not a prefixpoint, then $(x_i)_{i \leq \beta+1}$ with $x_{\beta+1} = x_\beta \vee O(x_\beta)$ is a strict extension of $\mathcal{N}$. □

Proposition 3.6. *If O is monotone, all monotone inductions of O (see Definition 2.1) are O-inductions in $\bot$ and vice versa.*

Proof. It is clear that all monotone inductions in $\bot$ are O-inductions since the definitions only differ in the second conditions, which is more restrictive for monotone inductions.

For the other direction, let $(x_i)_{i \leq \beta}$ be any O-induction. We first claim that each x_i is a postfixpoint of O. This claim is clear for $i = 0$ and the limit of an increasing sequence of postfixpoints of a monotone operator is a postfixpoint. If x_i is a postfixpoint, the $x_i \leq O(x_i)$ and hence $O(x_i) \vee x_i = O(x_i)$. Furthermore, since O is monotone, $O(x_i) \leq O(x_{i+1})$. Combining these two equations we get that

$$x_i \leq x_{i+1} \leq O(x_i) \vee x_i = O(x_i) \leq O(x_{i+1})$$

and see that indeed, x_{i+1} is a postfixpoint as well.

Hence, for each i, $x_i \leq O(x_i)$ and thus $x_i \vee O(x_i) = O(x_i)$. Thus, $(x_i)_{i \leq \beta}$ is indeed a monotone induction, which we needed to show. □

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

$$\left\{ \begin{array}{l} p \\ q \leftarrow p \\ r \leftarrow s \\ r \leftarrow p \end{array} \right\}$$

This is a positive logic program, hence $T_\mathcal{P}$ is monotone. The following are the three terminal strict $T_\mathcal{P}$-inductions in $\bot = \emptyset$:

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

They indeed all converge to the intended model of $\mathcal{P}$, namely the least fixpoint of $T_\mathcal{P}$. ▲

Example 3.9. Let $\mathcal{P}$ be the logic program

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

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 $\bot = \emptyset$.

$$\begin{aligned} \mathcal{N}_1 &= (\emptyset, \{q\}, \{p, q\}) \\ \mathcal{N}_2 &= (\emptyset, \{p, q\}) \\ \mathcal{N}_3 &= (\emptyset, \{p\}) \end{aligned}$$ ▲

The previous example shows that certain derivations in an O-induction can 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 these two sequences, the rule was applied when it was not *safe* to do so. Below, we formally define a notion of safety to avoid such premature derivations, i.e., to only derive facts that remain derivable, regardless of which other derivations are made further on in the induction process.

Definition 3.10. 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., $x' \leq x \vee O(x_\beta)$.

Proposition 3.11. *If y is safely derivable from x and $x \leq z \leq y$, then z is safely derivable from x.*

Proof. It follows directly from the definition that z is derivable from x. To see that it is safely derivable, take any O-induction in x with limit x_β. Then $z \leq y \leq x \vee O(x_\beta)$ and the result follows. □

Proposition 3.12. *If $Y \subseteq L$ and each $y \in Y$ is safely derivable from x, then $\bigvee Y$ is safely derivable from x. Hence, for each x, there exists a largest y such that y is safely derivable from x.*

Proof. It is easy to see that $\bigvee Y$ is derivable from x. To see that it is safely derivable, take any O-induction in x with limit x_β. Then $y \leq x \vee O(x_\beta)$ for each $y \in Y$ and hence also $\bigvee Y \leq x \vee O(x_\beta)$ and the result follows. □

Example 3.13 (Example 3.8 continued)**.** All derivations in all of the inductions here are safe. Consider for instance the derivation of $\{p, r\}$ from $\{p\}$ in $\mathcal{N}_2$. For each interpretation $I \supset \{p\}$, it holds that $r \in T_\mathcal{P}(I)$. Hence, for each $T_\mathcal{P}$ induction $(x_i)_{i \leq \beta}$ in $\{p\}$ it holds that $x_\beta \supset \{p\}$ and thus that $r \in T_\mathcal{P}(x_\beta)$. Now this means that $\{p, r\} \subseteq \{p\} \cup T_\mathcal{P}(x_\beta)$ and thus indeed, this derivation is safe. ▲

The situation in Example 3.13 is not a coincidence, as the following proposition shows.

Proposition 3.14. *If O is monotone and y is derivable from x, then y is safely derivable from x.*

Proof. Suppose y is derivable from x, i.e., that $x \leq y \leq O(x) \vee x$. Let $(x_i)_{i \leq \beta}$ be any O-induction in x. Then $x_\beta \geq x$ and hence, by monotonicity of O, $O(x_\beta) \geq O(x)$. Thus, indeed $y \leq x \vee O(x) \leq x \vee O(x_\beta)$, as we needed to show. □

For non-monotone operators, the situation is different, as is to be expected.

Example 3.15 (Example 3.9 continued)**.** The (intuitively) wrong derivation of $\{q\}$ from $\emptyset$ is not safe. Indeed, $\mathcal{N}_1$ is a $T_\mathcal{P}$-induction with as limit $\{p, q\}$, but

$$\{q\} \not\subseteq \emptyset \cup T_\mathcal{P}(\{p, q\}) = \emptyset \cup \{p\} = \{p\}.$$ ▲

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

Definition 3.16. 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 3.9, we showed that not all terminal O-inductions converge to the same lattice point. Luckily, the safety criterion leads to a better situation.

Theorem 3.17. *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 3.18. *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. The product order $\leq$ for ordinal pairs (given by $(i,j) \leq (k,l)$ if $i \leq k, j \leq l$) is a well-founded order, hence every set of such pairs contains minimal elements in this order.

Assume towards contradiction that pairs $(i,j) \leq (\beta,\gamma)$ exist that contradict this lemma, and let (i,j) be a minimal such pair in the product order. Hence, either $x_{i+1} \vee y_j$ exists and is not safely derivable from $x_i \vee y_j$, or $x_i \vee y_{j+1}$ exists and is not safely derivable from $x_i \vee y_j$.

Assume that it is the first case. Thus, $i+1 \leq \beta$ and $x_{i+1} \vee y_j$ is not safely derivable from $x_i \vee y_j$. By the minimality of (i,j), the sequence $\mathcal{N}" = (x_i \vee y_k)_{0\leq k\leq j}$ is a safe O-induction from x_i with limit $x_i \vee y_j$. Since $x_i \leq x_{i+1}$, also $x_i \vee y_j \leq x_{i+1} \vee y_j$. Since x_{i+1} is safely derivable from x_i, and $\mathcal{N}"$ is an O-induction with limit $x_i \vee y_j$, it holds that $x_{i+1} \leq x_i \vee O(x_i \vee y_j)$, hence also $x_{i+1} \vee y_j \leq x_i \vee y_j \vee O(x_i \vee y_j)$. Hence $x_{i+1} \vee y_j$ is derivable from $x_i \vee y_j$. We now show that this derivation is safe.

Since x_{i+1} is safely derivable from x_i, for each O-induction in x_i with limit z, it must hold that $x_{i+1} \leq x_i \vee O(z)$. Now, each O-induction in $x_i \vee y_j$ can be turned into an O-induction in x_i by composing $\mathcal{N}"$ with it, hence, for each O-induction from $x_i \vee y_j$ with limit z, it must also hold that $x_{i+1} \leq x_i \vee O(z)$, thus that $x_{i+1} \vee y_j \leq x_i \vee y_j \vee O(z)$. Thus, $x_{i+1} \vee y_j$ is safely derivable from $x_i \vee y_j$, which yields a contradiction.

The second case is obtained by a symmetrical argument. □

Proof of Theorem 3.17. 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 3.18, 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 3.19. The *safely defined point* by O, denoted $\mathit{safe}(O)$ is the limit of all safe-terminal O-inductions in $\bot$.

Example 3.20 (Example 3.9 continued)**.** The induction $\mathcal{N}_3$ is the only safe-terminal induction in $\emptyset$ in which each derivation is strict. Its limit is the intended model of $\mathcal{P}$, namely the perfect model. ▲

By Theorem 3.17, the safely defined point is well-defined. We now study some properties of the safely defined point.

Proposition 3.21. *For any operator O, $\mathit{safe}(O)$ is a postfixpoint of O, i.e., $\mathit{safe}(O) \leq O(\mathit{safe}(O))$.*

Proof. Let $\mathcal{N} = (x_i)_{i\leq\beta}$ be a safe O-induction in $\bot$. We show by induction that $x_i \leq O(x_\beta)$ for each i. The claim trivially holds for $i = 0$ since $x_0 = \bot$. It is preserved in limit ordinals i since $\bigvee_{j<i} x_j \leq O(x_\beta)$ if $x_j \leq O(x_\beta)$ for each $j < i$. We show that it also holds for successor ordinals. Hence, assume that $x_i \leq O(x_\beta)$ with $i < \beta$ and that x_{i+1} is safely derivable from x_i. Since $\mathcal{N}$ is an O-induction and x_{i+1} is safely derivable from x_i, it must hold that

$$x_{i+1} \leq x_i \vee O(x_\beta) \leq O(x_\beta) \vee O(x_\beta) = O(x_\beta)$$

and the result follows. □

Example 3.22. The safely defined point is not always a fixpoint of O. 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 3.23. We call an operator O *complete* if the safely defined point by O is a fixpoint of O, i.e., if $O(\mathit{safe}(O)) = \mathit{safe}(O)$.

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

Proposition 3.24. *An operator O is complete if and only if every safe-terminal O-induction in $\bot$ is terminal.*

Proof. If O is complete, $\mathit{safe}(O)$ is a fixpoint of O. It follows then from Proposition 3.5 that every O-induction with limit $\mathit{safe}(O)$ is terminal.

On the other hand, assume that every safe-terminal O-induction in $\bot$ is terminal. Thus (by Proposition 3.5) $\mathit{safe}(O)$ is a prefixpoint of O. By Proposition 3.21, $\mathit{safe}(O)$ is also a postfixpoint of O. Hence, it must be a fixpoint of O. □

Proposition 3.25. *If O is a monotone operator, then O is complete and $\mathit{safe}(O) = \mathrm{lfp}(O)$.*

Proof. From the monotonicity of O, it easily follows that every O-induction is safe. The result then follows from Corollary 3.7. □

Proposition 3.26. *If O is an anti-monotone operator, then $\mathit{safe}(O) = \mathrm{lfp}(O^2)$.*

Proof. Consider for any ordinal β the sequences $\mathcal{N} = (x_i)_{i \leq \beta}$ given by

$$\begin{aligned} x_0 &= \bot \\ x_{i+1} &= O(y_i), \text{ for each } i < \beta \\ x_\lambda &= \bigvee_{i<\lambda} x_i, \text{ for each limit ordinal } \lambda \leq \beta \end{aligned}$$

and $(y_i)_{i \leq \beta}$ given by

$$\begin{aligned} y_0 &= \top \\ y_{i+1} &= O(x_i), \text{ for each } i < \beta \\ y_\lambda &= \bigwedge_{i<\lambda} y_i, \text{ for each limit ordinal } \lambda \leq \beta \end{aligned}$$

We will prove the following claims about this sequence.

1. For each $i \leq \beta$, $x_i \leq y_i$ and if $i < \beta$, $x_i \leq x_{i+1}$ and $y_{i+1} \leq y_i$.
2. $(x_i)_{i \leq \beta}$ is a monotone O^2-induction.
3. $(x_i)_{i \leq \beta}$ is a safe O-induction.

The first statement follows from the construction of the two sequences and the fact they converge to the so-called *least alternating pair* of the anti-monotone operator O. We do prove this below for completeness.

From the last two claims it follows that for β large enough $x_\beta = \mathit{safe}(O^2) = \mathrm{lfp}(O^2)$. Since $O(x_\beta)$ is derivable from x_β and $O(O(x_\beta)) = x_\beta$, it follows that $(x_i)_{i \leq \beta}$, as an O-induction, is safe-terminal. Hence

$$\mathit{safe}(O) = x_\beta = \mathit{safe}(O^2) = \mathrm{lfp}(O^2),$$

which is what we needed to show. We now show that the claims we made indeed hold.

1. We prove this by induction.
   It certainly holds for $i = 0$ since $x_0 = \bot$ and $y_0 = \top$.
   If the claim holds for $i < \beta$, then $x_i \leq y_i$. Hence by anti-monotonicity of O,
   $$y_{i+1} = O(x_i) \geq O(y_i) = x_{i+1}.$$
   Also, since $i + 1 < \beta$ and the claim holds for i, it holds that $x_i \leq x_{i+1}$ and $y_{i+1} \leq y_i$. Now, by anti-monotonicity of O,
   $$y_{i+1} = O(x_i) \geq O(x_{i+1}) = y_{i+2}$$
   and
   $$x_{i+2} = O(y_{i+1}) \geq O(y_i) = x_{i+1}.$$
   Hence, we proved that all three inequalities in the claim also hold for $i + 1$.
   Finally, suppose the claim holds for all $i < \lambda$ with $\lambda \leq \beta$ some limit ordinal. For each $i \leq j < \lambda$, it holds that $x_i \leq x_j \leq y_j$. Similarly, for each $j \leq i < \lambda$, it holds that $x_i \leq y_i \leq y_j$. Hence, for all $i, j < \lambda$, $x_i \leq y_j$. Thus also
   $$x_\lambda = \bigvee_{i<\lambda} x_i \leq \bigwedge_{j<\lambda} y_j = y_\lambda.$$
   If furthermore $\lambda < \beta$, then we know that for each $j < \lambda$, $y_j \geq y_\lambda$. Hence, by anti-monotonicity of O, $x_{j+1} = O(y_j) \leq O(y_\lambda)$ for each $j < \lambda$. Thus
   $$x_\lambda \leq \bigwedge_{j<\lambda} x_{j+1} \leq O(y_\lambda) = x_{\lambda+1}.$$
   Similarly we find that also
   $$y_\lambda \geq y_{\lambda+1}$$
   and we see that the claim indeed also holds for λ.
   Hence, we showed by transfinite induction that the first claim is indeed satisfied for all $i \leq \beta$.
2. To see that $(x_i)_{i\leq\beta}$ is a monotone O^2-induction, we note that by the first claim, for each i:
   $$y_i \geq y_{i+1},$$
   hence by anti-monotonicity of O also
   $$x_{i+1} = O(y_i) \leq O(y_{i+1}) = O^2(x_i).$$
   Thus, it holds that
   $$x_i \leq x_{i+1} \leq O^2(x_i).$$
3. First, we show that for each $i < \beta$, x_{i+1} is derivable from x_i. To see this, note that from the first claim, it follows that
   $$x_i \leq x_{i+1} \leq y_{i+1} = O(x_i),$$
   hence x_{i+1} is indeed derivable from x_i.
   Second, we show that x_{i+1} is *safely* derivable from x_i. To show this, fix i and let $(z_j)_{j\leq\gamma}$ be any O-induction in x_i. We need to show that $x_{i+1} \leq x_i \vee O(z_\gamma)$. In order to show this, we claim that for each j, $x_i \leq z_j \leq y_i$. We show this claim by (transfinite) induction on j. It certainly holds for $j = 0$, since $z_0 = x_i$. If our claim holds for j, since z_{j+1} is a refinement of z_j, it holds that $z_j \leq z_{j+1} \leq z_j \vee O(z_j)$. By our induction hypothesis, and the fact that O is anti-monotone,
   $$x_i \leq z_j \leq z_{j+1} \leq z_j \vee O(z_j) \leq y_j \vee O(x_j) = y_j \vee y_{j+1} = y_j$$
   and indeed our claim follows. It is easy to see that it also holds in limit ordinals.
   Now, since our claim holds for all $j \leq \gamma$, it also holds for $j = \gamma$. Hence, we find that $y_i \geq z_\gamma$ and thus that $x_{i+1} = O(y_i) \leq O(z_\gamma)$ and it indeed follows that x_{i+1} is safely derivable from x_i, which is what we still needed to show. □

Theorem 3.17 shows that safe O-inductions, despite their high degree of 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, especially to those with a constructive characterization? We study this in Section 6. First, we study complexity of some inference tasks related to safe inductions.

4. Complexity

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.

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 all $x \in L$ 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.[1]

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 essentially 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 4.1. *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.*

Proof. Algorithm 1 contains a nondeterministic program to decide that y is *not* safely derivable from x. It takes as input $\langle L, O\rangle, x$, and y.

Algorithm 1 Nondeterministic algorithm to decide that y is *not* safely derivable from x by O.

```
s ← x
while true do
    if y ⋠ x ∨ O(s) then
        return true
    else if O(s) ≤ s then
        return false
    else
        choose an s′ ∈ L_i with s < s′ ≤ s ∨ O(s)
        s ← s′
    end if
end while
```

This program nondeterministically traverses an O-induction from x. Every state x' that can be reached by a natural induction from x can be reached by a run of this program. The algorithm stops with `true` when

[1] As you might notice, the input of some of the problems we consider also contains, besides $\langle L, O\rangle$ a number of lattice points. It is always possible to encode lattice points in a way that is polynomial in terms of the sum of the branching width and height of the lattice, e.g., by describing paths from $\bot$ to the given lattice point.

it reaches a lattice point that provides a counterexample for the safe derivability of y. It stops with `false` if the reached structure is a prefixpoint of O and it still provides no counterexample for the safe derivability of y. In this case, the traversed O-induction was a terminal one that was not a witness that y was not safely derivable.

One run of the algorithm builds a strictly growing sequence of lattice points; hence, the number of iterations is bound by the height of the lattice. At each step, the main computations are the computations of $O(x)$, $x \vee O(x)$ and checking for two lattice points whether one is smaller than the other. In order to branch on all elements $\{s' \mid s \leq s' \leq s \vee O(s)\}$, we can iteratively compute all direct successors of s and check whether they satisfy the condition above. These operations require to solve the function problem $F_{\mathcal{C}}$. Hence, Algorithm 1 runs in nondeterministic polynomial time that has a run and terminates with `true` if and only if y is not safely derivable from x. It follows that deciding that y is not safely derivable from x is in NP and its dual is in co-NP. □

Theorem 4.2. *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.*

Proof. We first show containment in Δ_2^P. First note that if y is safely derivable from x, then so is every z with $x \leq z \leq y$. Hence, the safely defined point can be reached by a safe O-induction such that for each j, x_{j+1} is a direct successor of x_j. By solving a polynomial number (in terms of the branching width of L) of co-NP problems, we can compute the set of direct successors y of x that are safely derivable from x. By doing this a polynomial number of times (in terms of the height of L), we find the safely defined structure and can determine whether $safe(O) \leq s$.

We now show the hardness result. To do this, we encode the co-NP hard problem of deciding validity of a propositional formula φ (over a propositional vocabulary Σ) in Disjunctive Normal Form (DNF). Let Val denote a symbol not in Σ and $\Sigma' = \Sigma \cup \{Val\}$. Consider the lattice $\langle L, \leq\rangle = \langle 2^{\Sigma'}, \subseteq\rangle$, i.e., elements of L are propositional interpretations of Σ'. Consider a logic program $\mathcal{P}$ over Σ' that consists of a rule

$$Val \leftarrow x_1 \wedge \cdots \wedge x_n$$

for each disjunct $x_1 \wedge \cdots \wedge x_n$ of φ and of a rule

$$x \leftarrow Val$$

for each x in Σ.

We claim that $Val \in safe(T_{\mathcal{P}})$ if and only if φ is valid.

To see this, note that for each interpretation I of Σ', $Val \in T_{\mathcal{P}}(I)$ if and only if $I \models \varphi$.

Now, if φ is false in $\emptyset$, then φ is not valid. In this case, $\emptyset$ is a fixpoint of $T_{\mathcal{P}}$ and hence $(\emptyset)$ is the unique $T_{\mathcal{P}}$-induction. We see that in this case, $Val \notin safe(T_{\mathcal{P}})$ and φ is not valid.

Otherwise, $\emptyset \models \varphi$. In this case $\{Val\}$ is derivable from $\emptyset$, but not necessarily safely derivable. If φ is valid, then $Val \in T_{\mathcal{P}}(I)$ for each I and hence $\{Val\} \leq \emptyset \vee T_{\mathcal{P}}(I)$ for each I. Thus in this case, $\{Val\}$ is indeed safely derivable. On the other hand, if φ is not valid, there exists an interpretation $I \subseteq \Sigma$ such that $I \not\models \varphi$. Now, in this case $Val \notin T_{\mathcal{P}}(I \cup \{Val\})$. Notice that $I \cup \{Val\}$ is derivable from $\{Val\}$. This means that $(\emptyset, \{Val\}, \{Val\} \cup I)$ is a $T_{\mathcal{P}}$-induction with $\{Val\} \not\leq \emptyset \vee T_{\mathcal{P}}(I \cup \{Val\})$, i.e., that $\{Val\}$ is not safely derivable from $\emptyset$.

We conclude that $safe(T_{\mathcal{P}})$ is greater than $\{Val\}$ if and only if φ is valid. Since deciding validity of a sentence in DNF is co-NP hard, we obtain the desired result. □

As can be seen, there still is a gap in the complexity results: we managed to prove containment in Δ_2^P but only co-NP hardness in Theorem 4.2. While we tried, we did not manage to close this gap and present this as a challenge to the community.

5. 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 $x \leq y$, 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 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 $x \in L$. It follows from this definition that 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 $x = \text{lfp}(A(\cdot,y)_1)$ 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 by Denecker and Vennekens (2007).

Definition 5.1. 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 5.2. 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_λ,y_λ)=$lub_{\leq_p}\{(x_i,y_i) | i < \lambda\}$ 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. In the context of logic programming, elements of the bilattice $\left(2^\Sigma\right)^2$ are pairs (I_1,I_2) of interpretations. Such a pair (I_1,I_2) corresponds to a four-valued interpretation $\mathcal{I}$ that interprets each atom as true (**t**), false (**f**), unknown (**u**) or inconsistent (**i**):

$$p^{\mathcal{I}} = \begin{cases} \mathbf{t} & \text{if } p \in I_1 \text{ and } p \in I_2 \\ \mathbf{u} & \text{if } p \in I_1 \text{ and } p \notin I_2 \\ \mathbf{f} & \text{if } p \notin I_1 \text{ and } p \notin I_2 \\ \mathbf{i} & \text{if } p \notin I_1 \text{ and } p \in I_2 \end{cases}$$

$A \wedge B$		B		
		t	**f**	**u**
A	**t**	**t**	**f**	**u**
	f	**f**	**f**	**f**
	u	**u**	**f**	**u**

$A \vee B$		B		
		t	**f**	**u**
A	**t**	**t**	**t**	**t**
	f	**t**	**f**	**u**
	u	**t**	**u**	**u**

		$\neg A$
A	**t**	**f**
	f	**t**
	u	**u**

Figure 1: The Kleene truth tables (Kleene, 1938).

Truth values are ordered by the truth order $\leq_t$ induced by $\mathbf{f} \leq_t \mathbf{u} \leq_t \mathbf{t}$, $\mathbf{f} \leq_t \mathbf{i} \leq_t \mathbf{t}$. The pair (I_1, I_2) approximates all interpretations I' with $I_1 \subseteq I' \subseteq I_2$. We often identify an interpretation I with the four-valued interpretation (I, I). We are mostly concerned with consistent (also called partial or three-valued) 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* (**t**) in $\mathcal{I}$, the atoms in $I_2 \setminus I_1$ are *unknown* (**u**) in $\mathcal{I}$ and the other atoms are *false* (**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 the Kleene truth tables (see Figure 1).

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, given by

$$p^{\psi_{\mathcal{P}}(\mathcal{I})} = \max_{\leq_t} \{ body(r)^{\mathcal{I}} \mid r \in \mathcal{P} \wedge head(r) = p \}$$

DMT (2000) showed that the well-founded fixpoint of $\Psi_{\mathcal{P}}$ is the well-founded model of $\mathcal{P}$ as defined by Van Gelder et al. and that $\Psi_{\mathcal{P}}$-stable fixpoints are exactly the stable models of $\mathcal{P}$ as defined by Gelfond and Lifschitz. In this case, the operator $\Psi_{\mathcal{P}}(\cdot, y)_1$ coincides with the immediate consequence operator of the Gelfond-Lifschitz reduct (Gelfond and Lifschitz, 1988).

6. Safe Inductions and AFT

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

Theorem 6.1. *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 6.2. *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. The proof is by induction on the length β of the well-founded induction.

Our claim trivially holds for $\beta = 0$ and it is easy to see that it is preserved in limit ordinals. Assume it holds for i, we show that it also holds for $i + 1$; we distinguish two cases.

First, assume that $(x_i, y_i) \leq_p (x_{i+1}, y_{i+1}) \leq_p A(x_i, y_i)$.

1. We show that x_{i+1} is safely derivable from x_i. Since every tuple in a well-founded induction is consistent (Denecker and Vennekens, 2007), $(x_i, y_i) \leq_p (x_i, x_i)$ and hence it holds that

   $$x_i \leq x_{i+1} \leq A(x_i, y_i)_1 \leq A(x_i, x_i)_1 \leq O(x_i),$$

   hence x_{i+1} is derivable from x_i. Furthermore, we know that for each O-induction $\mathcal{N} = (z_i)_{i \leq \alpha}$ with $z_0 = x_i$, it holds that

   $$x_i \leq z_\alpha \leq y_i.$$

   Since A is an approximator of O and $z_\alpha \in [x_i, y_i]$, it holds that

   $$x_{i+1} \leq A(x_i, y_i)_1 \leq O(z_\alpha) \leq x_i \vee O(z_\alpha).$$

   Since this holds for each O-induction, x_{i+1} is indeed safely derivable from x_i.

2. Let $(z_i)_{i\leq\alpha}$ be an O-induction in x_i. From the induction hypothesis, it follows that $z_\alpha \leq y_i$. We show that $z_\alpha \leq y_{i+1}$ by induction on α. Since well-founded inductions are consistent and increasing in precision, it holds that $x_i \leq x_{i+1} \leq y_{i+1} \leq y_i$, hence our claim holds for $\alpha = 0$ (since $z_0 = x_i$). It is clear that this property is preserved in limit ordinals. Assume it holds for $\alpha = j$, we show that it also holds for $j+1$. We have that $x_i \leq z_j \leq y_i$. Since A is an approximator of O,

$$O(z_j) \in A(z_j, z_j) \geq_p A(x_i, y_i) \geq_p (x_{i+1}, y_{i+1}).$$

Hence, it follows that $O(z_j) \leq y_{i+1}$. Thus also $z_{j+1} \leq z_j \vee O(z_j) \leq y_{i+1}$ and indeed, the claim follows.

Second, assume that $x_{i+1} = x_i$ and $A(x_i, y_{i+1})_2 \leq y_{i+1} \leq y_i$.

1. The first claim is trivial since $x_{i+1} = x_i$.
2. Let $(z_j)_{j\leq\alpha}$ be an O-induction in x_i. From the induction hypothesis, it follows that $z_\alpha \leq y_i$. We show that $z_\alpha \leq y_{i+1}$ by induction on α. Since well-founded inductions are consistent and increasing in precision, it holds that $x_i \leq x_{i+1} \leq y_{i+1} \leq y_i$, hence our claim holds for $\alpha = 0$ (since $z_0 = x_i$). It is clear that this property is preserved in limit ordinals. Assume it holds for $\alpha = j$, we show that it also holds for $j+1$. We have that $x_i \leq z_j \leq y_{i+1}$. Since A is an approximator of O,

$$O(z_j) \in A(z_j, z_j) \geq_p A(x_i, y_{i+1}).$$

Hence $O(z_j) \leq A(x_i, y_{i+1})_2 \leq y_{i+1}$. Since also $z_j \leq y_{i+1}$, it follows that $z_{j+1} \leq z_j \vee O(z_j) \leq y_{i+1}$ and indeed, the claim follows. □

Proof of Theorem 6.1. 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 6.2. Furthermore, by the second point of Proposition 6.2 it holds that any O-induction stays under y_β; hence $z \leq y_\beta$. □

Example 6.3 (Example 3.9 continued)**.** In this example, the well-founded model is $(\{p\}, \{p\})$, i.e., it is two-valued. It follows immediately from Theorem 6.1 that in this case $safe(T_\mathcal{P}) = \{p\}$ as well. ▲

Example 6.4. Consider a logic program

$$\mathcal{P} = \left\{ \begin{array}{l} p \\ q \leftarrow \neg r \wedge p \\ r \leftarrow \neg q \wedge p \end{array} \right\}$$

The $\Psi_\mathcal{P}$-well-founded model of $\mathcal{P}$ is $(\{p\}, \{p, q, r\})$. In this case, the only safe-terminal strict induction is

$$(\emptyset, \{p\}).$$

Indeed, from $\{p\}$, the interpretations $\{p, q\}$, $\{p, r\}$, and $\{p, q, r\}$ are derivable, but none of them is safely derivable. Hence, the operator $T_\mathcal{P}$ is not complete. ▲

Example 6.5. Consider a logic program

$$\mathcal{P} = \left\{ \begin{array}{l} p \leftarrow p \\ p \leftarrow \neg p \end{array} \right\}$$

In this case, the $\Psi_\mathcal{P}$-well-founded model of $\mathcal{P}$ is $(\emptyset, \{p\})$. This is not two-valued. The safely defined point by $T_\mathcal{P}$ is $\{p\}$ since

$$(\emptyset, \{p\})$$

is a safe-terminal $T_\mathcal{P}$ induction. Hence, the operator $T_\mathcal{P}$ is complete. ▲

Theorem 6.1 has several consequences.

Corollary 6.6. *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$(O) = x$.*

The converse of Corollary 6.6 does not hold: it can be the case that O is complete while the A-well-founded fixpoint is not exact. This can be seen, e.g., in Example 6.11.

Corollary 6.7. *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 6.8. *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 6.9. *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.*

Proof. If x is an A-stable fixpoint of O, then $x = \text{lfp}(A(\cdot, x)_1)$. Consider the sequence $(x_i)_{i \leq \alpha}$ given by

$$\begin{aligned} x_0 &= \bot, \\ x_{i+1} &= A(x_i, x)_1, \\ x_\lambda &= lub(\{x_i \mid i < \lambda\}), \text{ for limit ordinals } \lambda. \end{aligned}$$

If α is large enough, it holds that $x = x_\alpha$. We claim that $(x_i)_{i \leq \alpha}$ is an O-induction. First, since $A(\cdot, x)_1$ is monotone, $x_{i+1} \geq x_i$ for each i. Second, notice that for each i, $x_i \leq x$, hence $x_i \in [x_i, x]$ and thus $O(x_i) \in A(x_i, x)$, i.e., $O(x_i) \geq A(x_i, x)_1$. From this we conclude that $x_{i+1} \leq O(x_i) \leq O(x_i) \vee x_i$. Thus $(x_i)_{i \leq \alpha}$ is indeed an O-induction. It is terminal since x is a fixpoint of O. □

Example 6.10. Consider the logic program

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

It holds that $\{p\}$ is a stable model of $\mathcal{P}$ (i.e., 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. In fact, the example below shows that there exist safe inductions such that the limit is not A-stable for *any* approximator of O.

Example 6.11. Consider the logic program

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

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. ▲

While Theorem 6.1 shows that the relation between safe inductions and the well-founded fixpoint is strong, Example 6.11 shows that the connection with the other fixpoints defined in AFT is weaker.

7. 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.

7.1. 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"[2], 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 actual state of the 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*[3] 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

[2] Or, following DMT (2011): "My knowledge entails".

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

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 7.1. 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 7.2. 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 of Definition 7.2 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 7.3 (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. $(\Sigma_i)_{0\leq i\leq n}$. 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 7.4. 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 7.3; in this case we define $pm(\mathcal{T})$ as $\text{lfp}(D_n) \oplus Q_{n-1}$.

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 7.5. 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 7.6. 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.

7.2. 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 7.1 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 7.7. The *safely defined semantics* is given by $Q \models_{sd} \mathcal{T}$ if $Q = \mathit{safe}(D_{\mathcal{T}})$ 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 7.8 (Example 7.1 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 $Q_q \stackrel{\text{def}}{=} \{\{p,q\},\{q\},\{p,q,r\},\{q,r\}\}$, 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 show that the above example is not a coincidence, i.e., that it generalizes to the class of monotonically stratified theories.

Theorem 7.9. *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 $\mathit{safe}(D_{\mathcal{T}})$ is the perfect model of $\mathcal{T}$.*

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

Lemma 7.10. *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*

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

Proof. For every interpretation it holds that $I \in D_{\mathcal{T}}(Q)$ if and only if

$$Q, I \models \mathcal{T}.$$

Since $(\mathcal{T}_i)_{0\leq i\leq n}$ is a partition of $\mathcal{T}$, this condition is equivalent with

$$Q, I \models \mathcal{T}_i \text{ for each } i.$$

To evaluate whether $Q, I \models \mathcal{T}_i$, objective atoms are evaluated with respect to I, and modal atoms with respect to Q. Since all objective atoms in $\mathcal{T}_i$ are over Σ_i and all modal atoms in $\mathcal{T}_i$ are over Σ'_i, the previous condition is equivalent with

$$Q|_{\Sigma'_i}, I|_{\Sigma_i} \models \mathcal{T}_i \text{ for each } i.$$

Hence, for each i,

$$D_{\mathcal{T}}(Q)|_{\Sigma_i} = D_{\mathcal{T}_i}(Q|_{\Sigma'_i})|_{\Sigma_i}$$

and the result follows. □

Lemma 7.11. *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 $D_{\mathcal{T}}(Q_1)|_{\Sigma'_i} = D_{\mathcal{T}}(Q_2)|_{\Sigma'_i}$.*

Proof. This is proven as part of Theorem 6.3 by Bogaerts et al. (2016). □

Lemma 7.10 shows how $D_{\mathcal{T}}$ is composed from the various $D_{\mathcal{T}_i}$. Lemma 7.11 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.

Proof of Theorem 7.9. In this proof, we will use the following notation. If $\Sigma' \subseteq \Sigma$ and Q' is a possible world structure over Σ', we use $\lceil Q' \rceil$ to denote the $\leq_k$-least possible world structure Q over Σ such that $Q|_{\Sigma'} = Q'$. It is easy to see that

$$\lceil Q' \rceil = \bigwedge\{Q \mid Q|_{\Sigma'} = Q'\} = Q' \oplus \bot|_{\Sigma\setminus\Sigma'}.$$

Let $(\mathcal{T}_i)_{0\leq i\leq n}$ be a monotonic stratification of $\mathcal{T}$ with respect to $(\Sigma_i)_{0\leq i\leq n}$. Furthermore, let Σ'_i denote $\bigcup_{j\leq i}\Sigma_j$ for each i. We will prove the following claim by induction on i.

Claim: for each i, there exists a safe $D_{\mathcal{T}}$-induction $\mathcal{N} = (Q_j)_{j\leq\beta}$ in $\bot$ such that

$$Q_\beta = \lceil pm(\bigcup\{\mathcal{T}_k \mid k \leq i\})\rceil.$$

Taking $i = n$, the theorem easily follows from the claim. Indeed, the claim than yields that

$$Q_\beta = \lceil pm(\bigcup\{\mathcal{T}_k \mid k \leq n\})\rceil = \lceil pm(\mathcal{T})\rceil = pm(\mathcal{T}),$$

i.e., that there exists a safe induction that has $pm(\mathcal{T})$ as limit. Since the perfect model of $\mathcal{T}$ is always a fixpoint of $D_{\mathcal{T}}$, the aforementioned safe-induction is terminal and $D_{\mathcal{T}}$ is indeed complete.

We now show that the claim indeed holds. The claim is trivial for the empty theory ($i = -1$). Assuming it holds for $i < n$, we show that it holds as well for $i + 1$. Let $\mathcal{N} = (Q_j)_{j\leq\beta}$ be a safe $D_{\mathcal{T}}$-induction in $\bot$ such that

$$Q_\beta = \lceil pm(\bigcup\{\mathcal{T}_k \mid k \leq i\})\rceil.$$

Consider the sequence $(Q'_k)_{k\leq\alpha}$ given by

$$\begin{aligned} Q'_0 &= Q_\beta, \\ Q'_\lambda &= \bigvee_{k<\lambda} Q'_k \text{ for limit ordinals } \lambda < \alpha, \\ Q'_{k+1} &= \lceil D_{\mathcal{T}}(Q_k)|_{\Sigma'_{i+1}}\rceil \text{ for each } k < \alpha. \end{aligned}$$

From Lemma 7.10, it follows that

$$\begin{aligned} Q'_{k+1} &= \lceil D_{\mathcal{T}}(Q_k)|_{\Sigma'_{i+1}}\rceil \\ &= \lceil D_{\bigcup_{j\leq i}\mathcal{T}_j}(Q_k)|_{\Sigma'_i} \oplus D_{\mathcal{T}_{i+1}}(Q_k)|_{\Sigma_{i+1}}\rceil. \end{aligned}$$

Since $Q_\beta|_{\Sigma'_i}$ is the perfect model of $\bigcup_{j\leq i}\mathcal{T}_j$ we then find that

$$Q'_{k+1} = \lceil Q_\beta|_{\Sigma'_i} \oplus D_{\mathcal{T}_{i+1}}(Q_k)|_{\Sigma_{i+1}}\rceil.$$

Or, using D_i as defined in Proposition 7.3, we find that

$$Q'_{k+1} = \lceil Q_\beta|_{\Sigma'_i} \oplus D_{i+1}(Q_k|_{\Sigma_{i+1}})\rceil.$$

Hence, for sufficiently large α, we find that $Q'_\alpha = \lceil pm(\bigcup\{\mathcal{T}_k \mid k \leq i+1\})\rceil$. What remains to prove is that

1. $(Q'_k)_{k\leq\alpha}$ is a $D_\mathcal{T}$-induction in Q_β, and
2. $(Q'_k)_{k\leq\alpha}$ is safe.

The fact **(1)**, that it is a $D_\mathcal{T}$-induction in Q_β, follows easily from the fact that for every possible world structure Q, $\lceil Q|_{\Sigma'}\rceil \leq_k Q$. Hence,

$$\begin{aligned} Q'_{k+1} &= \lceil D_\mathcal{T}(Q_k)|_{\Sigma'_{i+1}}\rceil \\ &\leq_k D_\mathcal{T}(Q_k). \end{aligned}$$

To see that **(2)** holds, i.e., that $(Q'_k)_{k\leq\alpha}$ is a safe $D_\mathcal{T}$-induction, take any $D_\mathcal{T}$-induction $(Q''_l)_{l\leq\gamma}$ in Q'_k. First we claim that for each l, $Q''_l|_{\Sigma'_i} = Q_\beta$. This claim clearly holds for $j=0$ since $Q''_0 = Q'_k$. If it holds for l, from Lemma 7.11, it follows that $D_\mathcal{T}(Q''_l)|_{\Sigma_i} = Q_\beta$ as well. Hence, it also holds for $l+1$. Similarly, we find that $Q''_l|_{\Sigma_{i+1}} \geq_k Q'_k|_{\Sigma_{i+1}}$. Since each D_i as defined in Proposition 7.3 is a monotone operator, we thus find that $Q'_k|_{\Sigma'_{i+1}} \leq_k Q''_l|_{\Sigma'_{i+1}}$. Furthermore, since $Q'_k = \lceil Q'_k|_{\Sigma'_{i+1}}\rceil$, we find that $Q'_k \leq_k Q''_l$ for each l. Taking $l=\gamma$, we find that $(Q'_k)_{k\leq\alpha}$ is indeed safe. □

8. Safe Inductions and Argumentation Frameworks

Abstract argumentation frameworks (AFs) (Dung, 1995) are simple and abstract systems to deal with contentious information and draw conclusions from it. An AF is a directed graph where the nodes are arguments and the edges encode a notion of attack between arguments. In AFs, we are not interested in the actual content of arguments; this information is abstracted away. In spite of their conceptual simplicity, there exist many different semantics with different properties in terms of characterization, existence and uniqueness.

Recently, Strass (2013) showed that many of the existing semantics of AFs (and, as a generalization, also of abstract dialectical frameworks (ADFs) (Brewka and Woltran, 2010; Brewka et al., 2013)) can be obtained by a direct applications of AFT. In this section we use the aforementioned study to relate safe inductions to AFs.

An *abstract argumentation framework* Θ is a directed graph (A, R) in which the nodes A represent arguments and the edges in R represent attacks between arguments. We say that a *attacks* b if $(a,b) \in R$. A set $S \subseteq A$ *attacks* a if some $s \in S$ attacks a. A set $S \subseteq A$ *defends* a if it attacks all attackers of a. An *interpretation* of an AF $\Theta = (A, R)$ is a subset S of A. The intended meaning of such an interpretation is that all arguments in S are accepted (or believed) and all arguments not in S are rejected. Interpretations are ordered according to the acceptance relation: $S_1 \leq S_2$ iff $S_1 \subseteq S_2$, i.e., if S_2 accepts more arguments than S_1. There exist many different semantics of AFs which each define different sets of acceptable arguments according to different standards or intuitions. The major semantics for argumentation frameworks can be formulated using two operators: the *characteristic function* F_Θ, which maps an interpretation S to

$$F_\Theta(S) = \{a \in A \mid S \text{ defends } a\}$$

and the operator U_Θ (U stands for unattacked), which maps an interpretation S to

$$U_\Theta(S) = \{a \in A \mid a \text{ is not attacked by } S\}.$$

An interpretation S is *conflict-free* if it is a postfixpoint of U_Θ ($S \leq U_\Theta(S)$), i.e., if no argument in S is attacked by S. The characteristic function is a monotone operator; its least fixpoint is called the *grounded extension* of Θ. The operator U_Θ is an anti-monotone operator; its fixpoints are called *stable extensions* of Θ. Many more semantics, such as *admissible interpretations*, *complete extensions*, *semi-stable extensions*, *stage extensions* and *preferred extensions* (Dung, 1995) can be characterized using the above operators as well.

The following theorem shows that the grounded extensions as defined in argumentation theory have very close ties to safe inductions.

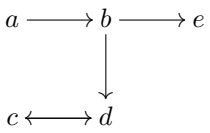
Theorem 8.1. *Let Θ be an argumentation framework. The following are all equal*

- *the grounded extension of Θ,*
- *the safely defined point by F_Θ,*
- *the safely defined point by U_Θ.*

Proof. Since the grounded extension of Θ is the least fixpoint of F_Θ, it follows from Proposition 3.25 that the first two are equal. To see that the last two are equal, we claim that U_Θ is an anti-monotone operator with $U^2_\Theta = F_\Theta$; the result then follows from Proposition 3.26.

To see that our claim holds, notice that $a \in F_\Theta(S)$ if and only if S defends a. This holds if and only if S attacks all attackers of a, i.e., if for all attackers b of a, $b \notin U_\Theta(S)$. This holds iff a is not a is not attacked by $U_\Theta(S)$, i.e., iff $a \in U^2_\Theta(S)$, which is what we needed to show. □

Example 8.2. Consider the following framework:



In this example a is unattacked, hence we expect it to be accepted; b is attacked by a, hence should not be accepted. The argument e is defended by a, hence can safely be accepted. c and d mutually attack each other and hence, defend themselves. Since we have already established that b is rejected, the only remaining argument that defends c is c itself. The grounded extension rejects self-defending arguments (i.e., rejects both c and d); it is $\{a, e\}$. The grounded extension is the limit of the following induction (that is both a safe F_Θ- and a safe U_Θ-induction:

$$(\emptyset, \{a\}, \{a, e\}).$$ ▲

9. 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 concepts in the context of autoepistemic logic, where we find that the safely defined point has interesting properties for a class of operators. We applied our theory to Dung's argumentation frameworks, where we found that for two existing operators, safe inductions yield the same (existing) semantics. It is a topic of future work to study the semantics induced by safe inductions for other application domains of AFT, such as abstract dialectical frameworks (Brewka and Woltran, 2010) and active integrity constraints (Flesca et al., 2004; Bogaerts and Cruz-Filipe, 2018). For the latter, 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).

Acknowledgements. This work was supported by the KU Leuven under project GOA 13/010 and by the Research Foundation - Flanders (FWO-Vlaanderen). Bart Bogaerts is a postdoctoral fellow of the Research Foundation – Flanders (FWO).

References

Antic, C., Eiter, T., Fink, M., 2013. Hex semantics via approximation fixpoint theory. In: Cabalar, P., Son, T. C. (Eds.), Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings. Vol. 8148 of LNCS. Springer, pp. 102–115.
URL http://dx.doi.org/10.1007/978-3-642-40564-8_11

Apt, K. R., Blair, H. A., Walker, A., 1988. Towards a theory of declarative knowledge. In: (Minker, 1988), pp. 89–148.

Bogaerts, B., Cruz-Filipe, L., 2018. Fixpoint semantics for active integrity constraints. Artif. Intell. 255, 43–70.
URL https://doi.org/10.1016/j.artint.2017.11.003

Bogaerts, B., Vennekens, J., Denecker, M., 2015. Grounded fixpoints and their applications in knowledge representation. Artif. Intell. 224, 51–71.
URL http://dx.doi.org/10.1016/j.artint.2015.03.006

Bogaerts, B., Vennekens, J., Denecker, M., Sep. 2016. On well-founded set-inductions and locally monotone operators. ACM Trans. Comput. Logic 17 (4), 27:1–27:32.
URL http://doi.acm.org/10.1145/2963096

Bogaerts, B., Vennekens, J., Denecker, M., 2017. Safe inductions: An algebraic study. In: (Sierra, 2017), pp. 859–865.
URL https://doi.org/10.24963/ijcai.2017/119

Brewka, G., Strass, H., Ellmauthaler, S., Wallner, J. P., Woltran, S., 2013. Abstract dialectical frameworks revisited. In: Rossi, F. (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. IJCAI/AAAI, pp. 803–809.
URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6551

Brewka, G., Woltran, S., 2010. Abstract dialectical frameworks. In: Lin, F., Sattler, U., Truszczyński, M. (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010. AAAI Press, pp. 102–111.
URL http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1294

Cruz-Filipe, L., Nov. 2016. Grounded fixpoints and active integrity constraints. In: Carro, M., King, A., De Vos, M., Saeedloei, N. (Eds.), Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, October 16-21, 2016, New York City, USA. Vol. 52 of OASIcs. Schloss Dagstuhl, pp. 11:1–11:14.
URL https://doi.org/10.4230/OASIcs.ICLP.2016.11

Denecker, M., Marek, V., Truszczyński, M., 2000. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In: Minker, J. (Ed.), Logic-Based Artificial Intelligence. Vol. 597 of The Springer International Series in Engineering and Computer Science. Springer US, pp. 127–144.
URL http://dx.doi.org/10.1007/978-1-4615-1567-8_6

Denecker, M., Marek, V., Truszczyński, M., 2003. Uniform semantic treatment of default and autoepistemic logics. Artif. Intell. 143 (1), 79–122.
URL http://dx.doi.org/10.1016/S0004-3702(02)00293-X

Denecker, M., Marek, V., Truszczyński, M., Jul. 2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. Information and Computation 192 (1), 84–121.
URL https://lirias.kuleuven.be/handle/123456789/124562

Denecker, M., Marek, V., Truszczyński, M., 2011. Reiter's default logic is a logic of autoepistemic reasoning and a good one, too. In: Brewka, G., Marek, V., Truszczyński, M. (Eds.), Nonmonotonic Reasoning – Essays Celebrating Its 30th Anniversary. College Publications, pp. 111–144.
URL http://arxiv.org/abs/1108.3278

Denecker, M., Vennekens, J., 2007. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In: Baral, C., Brewka, G., Schlipf, J. S. (Eds.), LPNMR. Vol. 4483 of Lecture Notes in Computer Science. Springer, pp. 84–96.
URL http://dx.doi.org/10.1007/978-3-540-72200-7_9

Denecker, M., Vennekens, J., 2014. The well-founded semantics is the principle of inductive definition, revisited. In: Baral, C., De Giacomo, G., Eiter, T. (Eds.), KR. AAAI Press, pp. 1–10.
URL http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7957

Denecker, M., Vennekens, J., Bogaerts, B., 2017. A logical study of some common principles of inductive definition and its implications for knowledge representation. CoRR abs/1702.04551.
URL http://arxiv.org/abs/1702.04551

Dung, P. M., 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77 (2), 321 – 357.
URL http://dx.doi.org/10.1016/0004-3702(94)00041-X

Fitting, M., 2002. Fixpoint semantics for logic programming — A survey. Theoretical Computer Science 278 (1-2), 25–51.
URL http://dx.doi.org/10.1016/S0304-3975(00)00330-3

Flesca, S., Greco, S., Zumpano, E., 2004. Active integrity constraints. In: Moggi, E., Warren, D. S. (Eds.), Proceedings of the 6th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, 24-26 August 2004, Verona, Italy. ACM, pp. 98–107.
URL http://doi.acm.org/10.1145/1013963.1013977

Gelfond, M., Lifschitz, V., 1988. The stable model semantics for logic programming. In: Kowalski, R. A., Bowen, K. A. (Eds.), ICLP/SLP. MIT Press, pp. 1070–1080.
URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.6050

Kleene, S. C., 1938. On notation for ordinal numbers. The Journal of Symbolic Logic 3 (4), 150–155.
URL http://www.jstor.org/stable/2267778

Levesque, H. J., 1990. All I know: A study in autoepistemic logic. Artif. Intell. 42 (2-3), 263–309.
URL http://dx.doi.org/10.1016/0004-3702(90)90056-6

Liu, F., Bi, Y., Chowdhury, M. S., You, J., Feng, Z., 2016. Flexible approximators for approximating fixpoint theory. In: Khoury, R., Drummond, C. (Eds.), Advances in Artificial Intelligence - 29th Canadian Conference on Artificial Intelligence, Canadian AI 2016, Victoria, BC, Canada, May 31 - June 3, 2016. Proceedings. Vol. 9673 of Lecture Notes in Computer Science. Springer, pp. 224–236.
URL http://dx.doi.org/10.1007/978-3-319-34111-8_28

Marek, V., Truszczyński, M., 1991. Autoepistemic logic. J. ACM 38 (3), 588–619.
URL http://dx.doi.org/10.1145/116825.116836

Minker, J. (Ed.), 1988. Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann.

Moore, R. C., 1985. Semantical considerations on nonmonotonic logic. Artif. Intell. 25 (1), 75–94.
URL http://dx.doi.org/10.1016/0004-3702(85)90042-6

Przymusinski, T. C., 1988. On the declarative semantics of deductive databases and logic programs. In: (Minker, 1988), pp. 193–216.

Reiter, R., 1980. A logic for default reasoning. Artif. Intell. 13 (1-2), 81–132.
URL http://dx.doi.org/10.1016/0004-3702(80)90014-4

Sierra, C. (Ed.), 2017. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org.
URL http://www.ijcai.org/Proceedings/2017/

Strass, H., 2013. Approximating operators and semantics for abstract dialectical frameworks. Artif. Intell. 205, 39–70.
URL http://dx.doi.org/10.1016/j.artint.2013.09.004

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

van Emden, M. H., Kowalski, R. A., 1976. The semantics of predicate logic as a programming language. J. ACM 23 (4), 733–742.
URL http://dx.doi.org/10.1145/321978.321991

Van Gelder, A., Ross, K. A., Schlipf, J. S., 1991. The well-founded semantics for general logic programs. J. ACM 38 (3), 620–650.
URL http://dx.doi.org/10.1145/116825.116838

Vennekens, J., Gilis, D., Denecker, M., 2006. Splitting an operator: Algebraic modularity results for logics with fixpoint semantics. ACM Trans. Comput. Log. 7 (4), 765–797.
URL http://dx.doi.org/10.1145/1182613.1189735