On Well-Founded Set-Inductions and Locally Monotone Operators

BART BOGAERTS, Department of Computer Science, KU Leuven
JOOST VENNEKENS, Department of Computer Science, Campus De Nayer, KU Leuven
MARC DENECKER, Department of Computer Science, KU Leuven

In the past, compelling arguments in favour of the well-founded semantics for autoepistemic logic have been presented. In this paper, we show that for certain classes of theories, this semantics fails to identify the unique intended model. We solve this problem by refining the well-founded semantics. We develop our work in approximation fixpoint theory, an abstract algebraical study of semantics of nonmonotonic logics. As such, our results also apply to logic programming, default logic, Dung's argumentation frameworks and abstract dialectical frameworks.

CCS Concepts:•**Computing methodologies** → **Knowledge representation and reasoning;** *Nonmonotonic, default reasoning and belief revision;* Logic programming and answer set programming;

Additional Key Words and Phrases: Approximation fixpoint theory, Autoepistemic logic, Lattice operator, Logic programming, Nonmonotonic reasoning, Well-founded semantics

ACM Reference Format:
Bart Bogaerts, Joost Vennekens and Marc Denecker. 2016. On Well-Founded Set-Inductions and Locally Monotone Operators. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 33 pages.
DOI: 000000.000000

1. INTRODUCTION

The informal and formal semantics of nonmonotonic logics such as logic programming and autoepistemic logic have been a topic of at moments intense debate for several decades. Many semantics have been defined for these logics. In several foundational papers, Denecker and coauthors argued for a constructive interpretation of these logics and that well-founded inductions provide the desired construction. They went as far as to suggest that for "sound" theories, the well-founded construction always is strong enough to construct the intended (exact) model, and that if the well-founded construction gets trapped in a non-exact state, this is sign of a semantical anomaly in the theory which should be considered to be flawed. A well-known example of a "flawed" program is the following

$$\{p \leftarrow \neg p.\}.$$

This claim was made and proven for logic programs viewed as inductive definitions [Denecker and Vennekens 2014]. Such a claim was also made for autoepistemic logic [Denecker et al. 2011]. However, no proof was given for that case. Shortly after the latter paper was published, it became clear that for autoepistemic logic, this claim was incorrect: the well-founded model construction is not nearly strong enough to always construct the intended model.

An earlier version of this work has appeared as a chapter in a PhD dissertation from KU Leuven [Bogaerts 2015].

This research was supported by project GOA 13/010 of the Research Fund KU Leuven and projects G.0489.10, G.0357.12, and G.0922.13 of the Research Foundation - Flanders.
Author's addresses: Celestijnenlaan 200A, 3001 Heverlee, Belgium

DOI: 000000.000000

Example 1.1. In 2011, Hanne Vlaeminck, then PhD student of Denecker, presented the following, so far unpublished example to show the weakness of the well-founded semantics of autoepistemic logic.

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, he does not know 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.

Vlaeminck discovered this example when investigating a translation from ordered epistemic logic (OEL) [Vlaeminck et al. 2012] to AEL. The above AEL theory was obtained by applying this translation to some OEL theory with the intended interpretation $\{\{p, q\}, \{q\}\}$. Thus, this example showed not only the translation was not equivalence preserving, but also the weakness of the well-founded semantics, something that her supervisors—who had just advocated the well-founded semantics—certainly had not expected.

We make some observations regarding Example 1.1.

(1) As we show in Section 3, the well-founded semantics fails to identify the intended model.
(2) We can, informally, construct the intended model following some sort of stratification (first reasoning on knowledge of p, next on knowledge of q and finally on knowledge of r).
(3) The semantic operator[1] $D_{\mathcal{T}}$ associated to $\mathcal{T}$ has a unique fixpoint, which is the intended model and which is *grounded* (in the terminology of Bogaerts et al. [2015])

Denecker, Marek and Truszczyński (from now on abbreviated DMT) [Denecker et al. 2011] have strongly argued in favour of a constructive semantics for AEL. While a constructive semantics has indeed important advantages, the above example shows that the constructive semantics of their choice, the well-founded semantics, is too weak for making the construction. An alternative way suggested by the last observation is to use the grounded fixpoint semantics for AEL instead. However, the grounded fixpoint semantics is non-constructive. In order to solve this discrepancy, in this paper we define a novel and stronger constructive semantics for AEL. We show that on a large class of theories (*monotonically stratified* theories, a class that generalises Example 1.1), our stronger construction is capable of constructing a unique model. This gives strong evidence that indeed, the presented construction fits better with the aims of DMT. Moreover, we will prove that for each monotonically stratified theory $\mathcal{T}$, the model we construct is the unique grounded fixpoint of the semantic operator $D_{\mathcal{T}}$, i.e., that observation (3) above is no coincidence.

We develop our theory in approximation fixpoint theory (AFT) [Denecker et al. 2000], an algebraical study of logics with a fixpoint semantics. DMT [2000] developed this theory after discovering analogies in the semantics of logic programming, AEL [Moore 1985] and default logic (DL) [Reiter 1980]. Their theory defines different types of fix-

[1] In the preliminaries, this operator and other related concepts are formally defined.

points (supported, Kripke-Kleene, stable, well-founded) for a so-called approximating operator. In the context of logic programming, they showed that Fitting's (three- or four-valued) immediate consequence operator is an approximator of the two-valued immediate consequence operator and that its different types of fixpoints correspond exactly with the major, equally named, semantics of logic programs. They identified approximating operators for default logic and autoepistemic logic and showed that AFT induces all main and some new semantics in these fields [Denecker et al. 2003]. Moreover, this work unified DL and AEL in a deep sense. They showed that Konolige's mapping from DL to AEL [Konolige 1988] preserves the approximating operator and hence, preserves all of the types of semantics. Thus, DL can (and should) be viewed as a fragment of AEL under Konolige's embedding and an old research question was resolved [Denecker et al. 2011]. Recently, Strass [2013] showed that also the major semantics of Dung's argumentation frameworks [Dung 1995] and abstract dialectical frameworks [Brewka and Woltran 2010] can be characterised using AFT. Nowadays, AFT is used for various purposes. These purposes include defining extensions of logic programming [Antic et al. 2013], defining new logics [Bogaerts et al. 2014], integrating different formalisms [Bi et al. 2014], studying complexity [Strass and Wallner 2014], and studying modularity and predicate introduction for logics with a fixpoint semantics in a uniform way [Vennekens et al. 2006; Vennekens et al. 2007b; 2007c]. Recently, Bogaerts et al. [2015] extended AFT with a new class of fixpoints, called *grounded fixpoints*.

In Section 4, we define—algebraically—a refinement of the well-founded semantics. Our refined semantics is constructive and differs from the well-founded semantics in the sense that instead of approximating lattice elements by intervals (bilattice elements), we use arbitrary sets of lattice elements. This yields an approximation space of greater resolution, i.e., in which more fine-grained approximations can be represented. As we will show, it allows us to overcome the inaccuracy of the well-founded semantics in case of Vlaeminck's problem. In Section 5, we introduce a class of lattice operators, called *locally monotone* operators. We prove that for locally monotone operators, our refined semantics yields a single fixpoint, the unique grounded fixpoint of the operator. In Section 6, we show that monotonically stratified AEL theories induce a locally monotone operator and that the unique fixpoint identified by our semantics coincides with the intended model for monotonically stratified theories. Thus, we show that the grounded fixpoint semantics is correct for monotonically stratified theories and provide a general, algebraical, construction of the unique grounded fixpoint.

In Section 7, we briefly apply our theory to logic programming. In this context, we show that (locally) stratified logic programs [Przymusinski 1988] induce a locally monotone operator.

In Section 8, we relate our theory to the work by Vennekens et al. [2006] on modularity of lattice operators and the work by Niemelä [1991] on a constructive semantics for AEL. We conclude in Section 9.

In this paper, we only apply our theory to autoepistemic logic and logic programming, even though it is also applicable, e.g., to default logic and abstract argumentation frameworks. We also restrict our attention (e.g., in Definition 5.19) to finite stratifications since the infinite case (n is an infinite ordinal) yields no new insights while requiring several technical details. In Appendix A, we show that all of our major results remain valid in the infinite case. In Appendix B, we give some proofs that are omitted in the text.

2. PRELIMINARIES: AFT

In this section, we recall the basics of lattice theory and approximation fixpoint theory.

A *complete lattice* $\langle L, \leq \rangle$ is a set L equipped with a partial order $\leq$ such that every $S \subseteq L$ has a least upper bound ($\bigwedge S$) and a greatest lower bound ($\bigvee S$). We use the notations $x \wedge y = \bigwedge\{x, y\}$ and $x \vee y = \bigvee\{x, y\}$. A subset X of L is *chain complete* if every *chain* (every totally ordered subset) of X has a least upper bound and a greatest lower bound in X. A complete lattice is chain complete and has a least element $\bot$ and a greatest element $\top$ [Markowsky 1976].

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

Definition 2.1 (*Monotone induction*). If O is a monotone operator, a *monotone induction* of O is a (possibly transfinite) sequence $(x_i)_{i \leq \alpha}$ such that

(1) $x_0 = \bot$,
(2) $x_i \leq x_{i+1} \leq O(x_i)$, and
(3) $x_\lambda = \mathrm{lub}(\{x_i \mid i < \lambda\})$, for limit ordinals $\lambda \leq \alpha$.

A monotone induction is *terminal* if there exists no $x_{\alpha+1} \neq x_\alpha$ such that $(x_i)_{i \leq \alpha+1}$ is a monotone induction.

All terminal monotone inductions of O converge to $\mathrm{lfp}(O)$.

A point $x \in L$ is *grounded* for O if $x \leq v$ for each v such that $O(x \wedge v) \leq v$ [Bogaerts et al. 2015], it is *ungrounded otherwise*.

Given a lattice L, AFT makes use of the bilattice L^2. We define *projections* 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. L^c denotes the set of consistent elements. 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 (u, v) is more precise than (x, y) if it approximates less elements, i.e., if $[u, v] \subseteq [x, y]$. If L is a complete lattice, then $\langle L^2, \leq_p \rangle$ is also a complete lattice. Slightly abusing notation, we often use a bilattice point (x, y) as the set $[x, y]$ and write for instance $z \in (x, y)$ to denote that $x \leq z \leq y$.

AFT studies fixpoints of lattice operators $O : L \to L$ through operators approximating O. An operator $A : L^c \to L^c$ is a consistent *approximator* of O if it is $\leq_p$-monotone and has the property that for all x, $O(x) \in A(x, x)$. In the original work by DMT [2000], approximators were assumed to be *exact* ($A(x, x) = (O(x), O(x))$ for all x), but all properties are preserved when weakening this condition to $O(x) \in A(x, x)$. This weakened form was introduced by Denecker and Vennekens [2007] since non-exact approximators sometimes naturally arise, for instance when computing the well-founded model of a definition in the context of a *partial* interpretation of the open symbols. AFT studies fixpoints of O using fixpoints of A as follows [Denecker et al. 2000]. The *A-Kripke-Kleene fixpoint* is the $\leq_p$-least fixpoint of A and has the property that it approximates all fixpoints of O. A *partial A-stable fixpoint* is a pair (x, y) such that $x = \mathrm{lfp}(A(\cdot, y)_1)$ and $y = \mathrm{lfp}(A(x, \cdot)_2)$, where $A(\cdot, y)_1$ denotes the operator $L \to L : x \mapsto A(x, y)_1$ and analogously for $A(x, \cdot)_2$. The *A-well-founded fixpoint* is the least precise partial A-stable fixpoint. An *A-stable fixpoint* of O is a fixpoint x of O such that (x, x) is a partial A-stable fixpoint. This is equivalent with the condition that $x = \mathrm{lfp}(A(\cdot, x)_1)$.

The A-Kripke-Kleene fixpoint of O can be constructed as the limit of any monotone induction of A. Denecker and Vennekens [2007] also worked out a constructive characterisation of the A-well-founded fixpoint.

Definition 2.2 (*A-refinement*). An *A-refinement* of (x, y) is a pair $(x', y') \in L^2$ satisfying one of the following two conditions:

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

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

The first type of refinements are called *application refinements* and the second type of refinements are *unfoundedness refinements*.

Definition 2.3 (*Well-founded induction*). A *well-founded induction* of A is a sequence $(x_i, y_i)_{i \leq \beta}$ with β an ordinal such that

(1) $(x_0, y_0) = (\bot, \top)$,
(2) (x_{i+1}, y_{i+1}) is an A-refinement of (x_i, y_i), for all $i < \beta$, and
(3) $(x_\lambda, y_\lambda) = \mathrm{lub}_{\leq_p} \{(x_i, y_i) \mid i < \lambda\}$ 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 well-founded inductions of A have the same limit, namely the A-well-founded fixpoint of O. Well-founded inductions generalise the well-founded model construction for logic programs [Van Gelder et al. 1991].

In general, a lattice operator $O : L \to L$ has a family of approximators. DMT [2004] showed that there is a *most precise*[2] approximator, U_O, called the *ultimate approximator* of O, defined by $U_O : L^c \to L^c : (x, y) \mapsto (\bigwedge O([x, y]), \bigvee O([x, y]))$. Here, we used the notation $O(X) = \{O(x) \mid x \in X\}$ for a set $X \subseteq L$. They showed that for every approximator A, all A-stable fixpoints are U_O-stable fixpoints, the U_O-Kripke-Kleene fixpoint is more precise than the A-Kripke-Kleene fixpoint and the U_O-well-founded fixpoint is more precise than the A-well-founded fixpoint. We refer to U_O-stable fixpoints as ultimate stable fixpoints of O, to the U_O-Kripke-Kleene fixpoints as the ultimate Kripke-Kleene fixpoint of O, and to the U_O-well-founded fixpoint as the ultimate well-founded fixpoint of O.

2.1. AFT and Logic Programming

We often illustrate our abstract results in the context of logic programming. We recall some preliminaries. We restrict ourselves to propositional logic programs, but allow arbitrary propositional formulas in rule bodies.[3] However, our results basically apply to all extensions of logic programming that admit an immediate consequence operator (non-propositional ones, aggregates in the body, etc.).

Let Σ be a propositional 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. A logic program $\mathcal{P}$ is a set of *rules* r of the form $h \leftarrow \varphi$, where h is an atom called the *head* of r, denoted $head(r)$, and φ is a propositional formula called the *body* of r, denoted $body(r)$. An interpretation I of the alphabet Σ is an element of 2^Σ, i.e., 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 based on the standard truth tables for propositional logic (see Figure 1).

With a logic program $\mathcal{P}$, we associate an immediate consequence operator [van Emden and Kowalski 1976] $T_\mathcal{P}$ that maps a structure I to

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

The *supported models* of $\mathcal{P}$ are the fixpoints of $T_\mathcal{P}$.

[2] For an exact definition and details, see [Denecker et al. 2004].
[3] This means that rule bodies can also contain material implication $\Rightarrow$, which is to be distinguished from the definitional implication (or *rule operator*) $\leftarrow$ that separates the head and body of a rule, see below.

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

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

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

Fig. 1. The truth tables for propositional logic.

$A \wedge B$		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**

Fig. 2. The Kleene truth tables [Kleene 1938].

For $p, q \in \Sigma$, we say that p *depends positively (respectively negatively)* on q (in $\mathcal{P}$) if q occurs in the scope of an even (respectively odd) number of negations in the body of some rule $r \in \mathcal{P}$ with $head(r) = p$. A logic program $\mathcal{P}$ is called *(locally) stratified*[4] [Apt et al. 1988; Przymusinski 1988] if every atom in Σ can be assigned an ordinal number (its rank) such that

— no atom depends on an atom with greater rank, and
— no atom depends negatively on an atom with the same rank.

In the context of logic programming, elements of the bilattice $\left(2^{\Sigma}\right)^2$ are four-valued interpretations, pairs $\mathcal{I} = (I_1, I_2)$ of interpretations. The pair (I_1, I_2) maps every atom to a tuple of two truth values; such a tuple corresponds to a four-valued truth value (true $(\mathbf{t}, \mathbf{t})$, false $(\mathbf{f}, \mathbf{f})$, unknown $(\mathbf{f}, \mathbf{t})$ or inconsistent $(\mathbf{t}, \mathbf{f})$). With this interpretation I_1 represents all atoms that are *certainly true* and I_2 represents all atoms that are *possibly true*. 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). If $\mathcal{I} = (I_1, I_2)$ is a (four-valued) interpretation, and $U \subseteq \Sigma$, we write $\mathcal{I}[U : \mathbf{f}]$ for the (four-valued) interpretation that equals $\mathcal{I}$ on all elements not in U and that interprets all elements in U as $\mathbf{f}$, i.e., the interpretation $(I_1 \setminus U, I_2 \setminus U)$. 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* ($\mathbf{t}$) in $\mathcal{I}$, the atoms in $I_2 \setminus I_1$ are *unknown* ($\mathbf{u}$) in $\mathcal{I}$ and the other atoms are *false* ($\mathbf{f}$) in $\mathcal{I}$. If $\mathcal{I}$ is a three-valued interpretation, and φ a formula, we write $\varphi^{\mathcal{I}}$ for the standard three-valued valuation based on the Kleene truth tables (see Figure 2).

Describing the semantics of a logic programs as fixpoints of a semantic operator has been a long-standing research topic. In early work, two-valued semantics of logic programs was studied in an operator-based fashion [Apt and van Emden 1982; van Emden and Kowalski 1976]. Later, this line of research has been generalised to stratified programs [Apt et al. 1988] and, to three-valued semantics [Fitting 1985; 1986]. In order to do this, Fitting defined the partial immediate consequence operator $\Psi_{\mathcal{P}}$ [Fitting 2002], a direct generalisation of $T_{\mathcal{P}}$ to partial interpretations as follows

$$\Psi_{\mathcal{P}}(\mathcal{I})_1 = \{a \in \Sigma \mid body(r)^{\mathcal{I}} = \mathbf{t} \text{ for some rule } r \in \mathcal{P} \text{ with } head(r) = a\},$$
$$\Psi_{\mathcal{P}}(\mathcal{I})_2 = \{a \in \Sigma \mid body(r)^{\mathcal{I}} \neq \mathbf{f} \text{ for some rule } r \in \mathcal{P} \text{ with } head(r) = a\}.$$

[4] Since we work in the propositional case, locally stratified logic programs and stratified logic programs are the same.

DMT [2000] showed that $\Psi_{\mathcal{P}}$ is an approximator of $T_{\mathcal{P}}$, that the well-founded fixpoint of $\Psi_{\mathcal{P}}$ is the well-founded model of $\mathcal{P}$ as defined by Van Gelder et al. [1991] and that $\Psi_{\mathcal{P}}$-stable fixpoints are exactly the stable models of $\mathcal{P}$ as defined by Gelfond and Lifschitz [1988]. 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]. The most precise approximator of $T_{\mathcal{P}}$ is the ultimate approximator $U_{\mathcal{P}}$.

2.2. AFT and Autoepistemic Logic

Autoepistemic logic (AEL) is a non-monotonic logic for modelling 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", 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, i.e., not in the scope of an operator K, in φ.

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 $\mathcal{W}_\Sigma$. 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 $\mathcal{L}_\Sigma$, 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 $I \in Q$, i.e., an agent knows φ if for each structure he deems possible (for each $I \in Q$), φ holds under the assumption that this structure *is* the real world. Moore [1985] associated with every theory $\mathcal{T}$ an operator $D_{\mathcal{T}}$ on the set of possible world structures as follows:

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

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

Following DMT [2003], the semantics of an autoepistemic theory is defined by an approximator on the bilattice $\mathcal{W}_\Sigma^2$. In this paper, we use the ultimate approximator $U_{\mathcal{T}}$ since this is the most precise approximator and we intend to show that even the ultimate well-founded model is not precise enough to capture the intended semantics of a class of autoepistemic theories. If we show this for the ultimate approximator, the same also trivially follows for all other approximators of $D_{\mathcal{T}}$. $U_{\mathcal{T}}$ maps (P, S) to (P', S'), where

$$P' = \{I \mid Q, I \models \mathcal{T} \text{ for some } Q \text{ with } P \supseteq Q \supseteq S\}$$
$$S' = \{I \mid Q, I \models \mathcal{T} \text{ for all } Q \text{ with } P \supseteq Q \supseteq S\}$$

The *(ultimate) well-founded model* of $\mathcal{T}$ is a belief pair, namely the $U_{\mathcal{T}}$-well-founded fixpoint.

DMT [2011] gave compelling arguments defending the (ultimate) well-founded semantics for autoepistemic reasoning. Well-founded inductions mimic the reasoning process of a rational agent: an (ultimate) well-founded induction $(P_i, Q_i)_{i\leq\beta}$ of $D_{\mathcal{T}}$ gradually derives what the agent knows (statements φ such that $P_i \models K\varphi$) and what it does not know (φ such that $Q_i \not\models K\varphi$).

Following Vennekens et al. [2006], we call an autoepistemic theory $\mathcal{T}$ *stratifiable*[5] 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 $Q_1 \oplus Q_2 \overset{\text{def}}{=} \{I_1 \cup I_2 \mid I_1 \in Q_1 \wedge I_2 \in Q_2\}$. If Q is a possible world structure over $\Sigma_1 \cup \Sigma_2$, the restriction of Q to Σ_1 is $Q|_{\Sigma_1} \overset{\text{def}}{=} \{I \cap \Sigma_1 \mid I \in Q\}$.

3. MOTIVATION

In this section, we first show that for certain AEL theories such as the theory in Example 1.1, the well-founded fixpoint is not precise enough. Afterwards, we generalise this example to obtain a class (called monotonically stratified theories) of theories for which we can unambiguously define the intended model. By analogy with a similar concept in logic programming, namely local stratification [Przymusinski 1988], we call the intended model of a monotonically stratified theory its *perfect model*. This perfect model is always a possible world structure (or, equivalently, an exact belief pair).

Example 3.1 (*Example 1.1 continued*). Again, let $\mathcal{T}$ be the autoepistemic theory

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

We now show that the well-founded semantics fails to compute the intended model $\{\{p, q\}, \{q\}\}$. Let $U_{\mathcal{T}}$ be the ultimate approximator of $D_{\mathcal{T}}$. First, we notice that $U_{\mathcal{T}}(\bot, \top) = (\bot, \top)$ and there are no strict refinements of the first kind of $(\bot, \top)$. Suppose that for some y', $U_{\mathcal{T}}(\bot, y')_2 \leq y'$. Since $D_{\mathcal{T}}(\bot) = \{\{q, r\}, \{q, r, p\}\}$, it follows that $y' \geq \{\{q, r\}, \{q, r, p\}\}$. We find that

$$\begin{aligned} y' &\geq U_{\mathcal{T}}(\bot, y')_2 \\ &= \bigvee\{D_{\mathcal{T}}([\bot, y'])\} \\ &\geq D_{\mathcal{T}}(\bot) \vee D_{\mathcal{T}}(\{\{q, r\}, \{q, r, p\}\}) \\ &\geq \{\{q, r\}, \{q, r, p\}\} \vee \{\{q\}, \{q, p\}\} \\ &= \emptyset = \top. \end{aligned}$$

Thus, there are also no strict refinements of the second kind either; $(\bot, \top)$ is the ultimate well-founded fixpoint of $D_{\mathcal{T}}$.

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

We observe that the theory from the above example is stratifiable with respect to the partition $\langle\{p\},\{q\},\{r\}\rangle$. Symbols in a stratum are uniquely defined in terms of knowledge of symbols in lower strata, hence we expect that there is a unique two-valued model which can be constructed *from the ground up*, following the stratification. This is not the case under the (ultimate) well-founded semantics. We now generalise this example.

Definition 3.2 (*Monotonically stratified*). 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 modal 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 stratification is *strict* if there are no modal subformulas $K\psi$ of $\mathcal{T}_i$ with $At(\psi)\subseteq\Sigma_i$.

The intuition regarding this definition is that whether or not symbols in Σ_i are known by the agent is completely determined by knowledge of symbols in $\Sigma_{i'}$ with $i' < i$ and by $\mathcal{T}_i$. Hence, knowledge of the agent can be built inductively via strata. Restricted to $\mathcal{T}_i$, symbols in Σ_i only depend positively on their own knowledge. This guarantees that $D_{\mathcal{T}}$ induces a monotonic operator when restricted to Σ_i.

PROPOSITION 3.3. *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.*

PROOF. Follows from the fact that all modal subformulas $K\psi$ of $\mathcal{T}_i$ with $At(\psi)\subseteq\bigcup_{0\leq j<i}\Sigma_j$ are evaluated with respect to Q_{i-1}, which is fixed, and all other modal formulas occur negatively, i.e., more knowledge can only yield more derivations. □

Proposition 3.3 can be used to iteratively build up a possible world structure similar to the perfect model for logic programs [Przymusinski 1988]. Given this similarity, we also call this constructed model *perfect*. Contrary to the case for logic programming, the well-founded model does not always equal the perfect model here, as Example 3.1 shows.

Definition 3.4 (*Perfect model*). 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 3.3; in this case we define $pm(\mathcal{T})$ as $\mathrm{lfp}(D_n)\oplus Q_{n-1}$.

Example 3.5 (*Example 3.1 continued*). Again, let $\mathcal{T}$ be the autoepistemic theory

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

This theory is monotonically stratified with respect the the partition $\langle\{p\},\{q\},\{r\}\rangle$. The perfect model of $\mathcal{T}$ is computed as follows.

$$\begin{aligned}
pm(\mathcal{T}_0) &= \{\emptyset,\{p\}\}\\
pm(\mathcal{T}_0\cup\mathcal{T}_1) &= \{\{q\}\}\oplus pm(\mathcal{T}_0) = \{\{q\},\{q,p\}\}\\
pm(\mathcal{T}_0\cup\mathcal{T}_1\cup\mathcal{T}_2) &= \{\emptyset\}\oplus pm(\mathcal{T}_1) = \{\{q\},\{q,p\}\}
\end{aligned}$$

where

$$\mathcal{T}_0 = \{\}$$
$$\mathcal{T}_1 = \{q \Leftrightarrow \neg Kp\}$$
$$\mathcal{T}_2 = \{r \Leftrightarrow \neg Kq\}$$

Hence, in this example, the perfect model indeed corresponds to the intended model. Furthermore, the perfect model computation follows the reasoning process described in Example 1.1 and it can be seen that $\{\{q\}, \{q, p\}\}$ is the unique grounded fixpoint of $D_{\mathcal{T}}$.

We shall show below in Corollary 6.5 that the perfect model of every "good" theory is the unique grounded fixpoint of its semantic operator.

The idea underlying the construction of the perfect model is that the knowledge of symbols in Σ_i remains fixed after computing the perfect model of $\bigcup_{j<i} \mathcal{T}_j$. Even though this seems intuitively clear, it is not always the case, as the following example shows.

Example 3.6. Let $\mathcal{T}$ be the AEL theory

$$\{p, Kp \Rightarrow q, Kp \Rightarrow \neg q\}.$$

This theory is monotonically stratified with respect to the partition $\langle\{p\}, \{q\}\rangle$. The lowest stratum determines the knowledge of p; it consists of the theory $\{p\}$, hence we conclude that the intended model Q of $\mathcal{T}$ satisfies $Q \models Kp$ and $Q \not\models K\neg p$. However, in the second stratum, an inconsistency occurs, since both q and $\neg q$ are derived. Hence the perfect model of $\mathcal{T}$ is the inconsistent possible world structure $\top = \emptyset$. Thus, in the perfect model of $\mathcal{T}$, also $\neg p$ is known, which violates our previous conclusion.

Luckily enough, this strange behaviour only occurs when the inconsistent possible world structure is involved. This kind of peculiarities has already been noted by Vennekens et al. [2007a]. They introduced the syntactical notion of a *permaconsistent* theory in order to avoid problems with the inconsistent possible world structure. We use a slightly weaker variant.

Definition 3.7 (*Weakly permaconsistent autoepistemic theory*). An autoepistemic theory $\mathcal{T}$ is called *weakly permaconsistent* if for every possible world structure Q, there is at least one I such that

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

PROPOSITION 3.8. *Suppose $\mathcal{T}$ is weakly permaconsistent and $(\mathcal{T}_i)_{0\leq i\leq n}$ is a monotonic stratification of $\mathcal{T}$ with respect to the partition $(\Sigma_i)_{0\leq i\leq n}$. Then, for every objective formula φ over $\bigcup_{j\leq i} \Sigma_j$, $pm(\mathcal{T}) \models K\varphi$ if and only if $pm(\bigcup_{j\leq i} \mathcal{T}_j) \models K\varphi$.*

PROOF. First observe that if Σ_1 and Σ_2 are two disjoint vocabularies, φ is a formula over Σ_1, and Q_1 and Q_2 are *non-empty* possible world structures over Σ_1 and Σ_2 respectively, then $Q_1 \models K\varphi$ if and only if $Q_1 \oplus Q_2 \models K\varphi$.

Now, if $\mathcal{T}$ is weakly permaconsistent, $D_{\mathcal{T}_i}(Q)$ is non-empty for all Q and i, i.e., all possible world structures used in the perfect model computation are non-empty. The result then follows from the above observation by induction. □

For "reasonable" theories (theories that represent the knowledge of a rational introspective agent), the inconsistent possible world structure will never be a relevant possible world structure. In this paper, we focus on the class of weakly permaconsistent theories, i.e., those in which consistency is guaranteed. What we expect is that a good semantics for AEL manages to identify the intended possible world structure for "reasonable" monotonically stratified theories.

Definition 3.9 (*Respecting stratification*). 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.

Following the analogy with logic programming, we would expect that stable and well-founded semantics respect stratification. However, this is not the case here! In fact, the motivation at the beginning of this section shows that well-founded semantics does not respect stratification. For (ultimate) stable semantics, this remains an open research question.

THEOREM 3.10. *The (ultimate) well-founded semantics for autoepistemic logic does not respect stratification.*

PROOF. Example 3.1 provides a counterexample for the ultimate well-founded semantics. For the well-founded semantics based on any other approximator A, for example the one defined in [Denecker et al. 2003], this follows from the algebraical property that the ultimate well-founded fixpoint is always more precise than the A-well-founded fixpoint. □

Intuitively, the problem with the well-founded semantics is the fact that points in the bilattice, which are intervals of lattice elements, are not granular enough to represent the information in the informal reasoning process presented in Example 1.1. There, we would first like to restrict our attention to the set of possible world structures in which both $\neg Kp$ and $\neg K\neg p$ hold, i.e., no information about p is known. However, the smallest (most precise) interval that approximates this set is $[\bot, \top]$; for this reason, well-founded inductions cannot leave the least precise bilattice point. In what follows, we define a refinement of the well-founded semantics that works on sets of lattice elements instead of intervals. This will allow us to follow the reasoning process of a rational agent in greater detail.

4. SET-INDUCTIONS

Given an operator $O : L \to L$, we now define two constructive characterisations of sets of elements of L and discuss how they relate to AFT. The first, and simplest, refines the Kripke-Kleene fixpoint. The Kripke-Kleene set, as we call it, manages to identify the intended model in Example 3.1 and for all monotonically stratified theories in which the stratification is strict. However, the semantics derived from it does not respect stratification in general. In order to overcome this limitation, the second constructive characterisation refines the well-founded semantics.

4.1. The Kripke-Kleene Set

We extend O to subsets of L in the pointwise manner. Below, the symbols X, X_i and Y denote subsets of L. Before defining our constructive characterisations, we recall how the ultimate Kripke-Kleene fixpoint is characterised. Given a set $X \subseteq L$, $[\![X]\!]$ denotes the smallest interval such that $X \subseteq [\![X]\!]$, i.e., $[\![X]\!] = [\bigwedge X, \bigvee X]$.

LEMMA 4.1. *Let O be an operator. (x', y') is an U_O-application refinement of (x, y) if and only if*

$$[\![O([x, y])]\!] \subseteq [x', y'] \subseteq [x, y].$$

PROOF. Trivial, since $U_O(x, y) = (\bigwedge O([x, y]), \bigvee O([x, y])) = [\![O([x, y])]\!]$. □

Now, the U_O-Kripke-Kleene fixpoint is characterised as the limit of any terminal monotone induction of U_O. In this computation, an interval $[x_i, y_i]$ is used to approximate the intended partial fixpoint. At each step, this interval is updated to the set

$S_i := O([x_i, y_i])$, and then for the next step in the induction, the smallest interval approximating S_i is taken. With this information, it is clear how to generalise the Kripke-Kleene fixpoint to sets: all we need to do is skip the approximation by intervals, which might lose valuable information.

Definition 4.2 (*Kripke-Kleene set-induction*). Let $O : L \to L$ be a lattice operator. A *Kripke-Kleene set-induction of* O, is a sequence $(X_i)_{i \leq \alpha}$ satisfying

— $X_0 = L$,
— $X_i \subseteq X_{i+1} \subseteq O(X_i)$, and
— $X_\lambda = \bigcap\{X_i \mid i < \lambda\}$, for limit ordinals $\lambda \leq \alpha$.

A Kripke-Kleene set-induction is *terminal* if there exists no $X_{\alpha+1} \neq X_\alpha$ such that $(X_i)_{i \leq \alpha+1}$ is a Kripke-Kleene set-induction induction.

LEMMA 4.3. *All terminal Kripke-Kleene set-inductions converge to the same set.*

PROOF. Follows immediately from the fact that Kripke-Kleene set-inductions are monotone inductions of the extension of O to 2^L. □

Definition 4.4 (*Kripke-Kleene set*). We call the unique set to which all terminal Kripke-Kleene set-inductions converge the *Kripke-Kleene set of* O and denote it $kks(O)$.

PROPOSITION 4.5. *Let* A *be an approximator of* O *and* (x, y) *be the* A*-Kripke-Kleene fixpoint of* O*. It then holds that* $kks(O) \subseteq [x, y]$.

PROOF. Follows immediately from Lemma 4.1 and the definition of $kks(O)$. □

PROPOSITION 4.6. *The Kripke-Kleene set of* O *contains all fixpoints of* O.

PROOF. Follows by induction from the fact that if X_i contains a fixpoint x of O, that then $O(X_i)$ also contains this fixpoint. □

The previous two propositions show that the Kripke-Kleene set refines the Kripke-Kleene fixpoint (for every approximator), but still approximates all fixpoints of O. For the example that motivated this work, the Kripke-Kleene set turns out to be precise enough.

Example 4.7 (*Example 3.1 continued*). Again take $\mathcal{T} = \{q \Leftrightarrow \neg Kp, r \Leftrightarrow \neg Kq\}$. The construction of the Kripke-Kleene set of $D_\mathcal{T}$ starts from the entire lattice $\mathcal{W}_\Sigma$. We make two observations.

(1) For every possible world structure Q, $D_\mathcal{T}(Q) \models K\neg q$ if $Q \models Kp$ and $D_\mathcal{T}(Q) \models Kq$ otherwise, and analogously for Kr.
(2) As p does not occur objectively in $\mathcal{T}$, all possible world structures Q in the image of $D_\mathcal{T}$ satisfy that $I \in Q$ iff $I \cup \{p\} \in Q$.

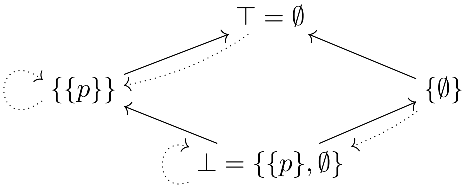
The above observations guarantee that $D_\mathcal{T}(\mathcal{W}_\Sigma) = \{Q, Q_q, Q_r, Q_{qr}\}$ where

— $Q \overset{\text{def}}{=} \{\{p\}, \{\}\}$,
— $Q_q \overset{\text{def}}{=} \{\{p, q\}, \{q\}\}$,
— $Q_r \overset{\text{def}}{=} \{\{p, r\}, \{r\}\}$, and
— $Q_{qr} \overset{\text{def}}{=} \{\{p, q, r\}, \{q, r\}\}$.

Furthermore, $D_\mathcal{T}(Q) = Q_{qr}$, $D_\mathcal{T}(Q_q) = Q_q$, $D_\mathcal{T}(Q_r) = Q_{qr}$, and $D_\mathcal{T}(Q_{qr}) = Q_q$. Thus $D^2_\mathcal{T}(\mathcal{W}_\Sigma) = \{Q_q, Q_{qr}\}$ and $D^3_\mathcal{T}(\mathcal{W}_\Sigma) = \{Q_q\}$. Thus, the Kripke-Kleene set of $D_\mathcal{T}$ is a singleton and the unique element of $kks(D_\mathcal{T})$ is the perfect model of $\mathcal{T}$.

In the next section we will show that it is not a coincidence that this theory has a singleton Kripke-Kleene set. A monotonically stratified theory in which the stratification is *strict* always has a singleton Kripke-Kleene set. This result does not hold for arbitrary monotonically stratified theories (if the stratification is not strict).

Example 4.8. The theory $\mathcal{T} = \{Kp \Rightarrow p\}$ is monotonically stratified with perfect model $\bot = \{\{p\}, \emptyset\}$. Its semantic operator is visualised below.



However, the Kripke-Kleene set of $D_{\mathcal{T}}$ is $\{\bot, \{\{p\}\}\}$.

4.2. The Well-Founded Set

In the previous section, we found that the Kripke-Kleene set sometimes is not precise enough. The reason why it fails is because it has no means to eliminate ungrounded reasoning. E.g., in Example 4.8, it fails to discover that knowledge of p only follows from the theory $Kp \Rightarrow p$ if p is known in the first place. Hence, a rational agent whose only knowledge is $\mathcal{T}$ would never know p. The well-founded semantics has a tool to eliminate this kind of reasoning, namely unfoundedness refinements. We generalise this kind of refinements to set inductions, but first, we recall some properties of ultimate well-founded inductions.

LEMMA 4.9. *Let $O : L \to L$ be an operator and U_O its ultimate approximator. Suppose $(x_i, y_i)_{i \leq \beta}$ is an U_O-well-founded induction. For every $i < \beta$, one of the following conditions is satisfied:*

1. $[\![O([x_i, y_i])]\!] \subseteq [x_{i+1}, y_{i+1}] \subseteq [x_i, y_i]$*, or*
2. $[x_i, y_i] \setminus [x_{i+1}, y_{i+1}]$ *is a set of ungrounded points and* $O([x_{i+1}, y_{i+1}]) \subseteq [x_{i+1}, y_{i+1}]$.

PROOF. If $(x_i, y_i) \to (x_{i+1}, y_{i+1})$ is an U_O-application refinement ((1) in Definition 2.2), then

$$\begin{aligned}[\![O([x_i, y_i])]\!] &= (\bigwedge O([x_i, y_i]), \bigvee O([x_i, y_i])) \\ &= U_O(x_i, y_i) \\ &\subseteq [x_{i+1}, y_{i+1}] \\ &\subseteq [x_i, y_i].\end{aligned}$$

In this case (1) is satisfied.

On the other hand, if $(x_i, y_i) \to (x_{i+1}, y_{i+1})$ is an unfoundedness refinement, then it has been shown by Bogaerts et al. [2015] (Theorem 4.6) that no grounded elements are removed. Furthermore, Denecker and Vennekens [2007] (Proposition 2) showed that it always holds that $U_O(x_{i+1}, y_{i+1}) \subseteq [x_{i+1}, y_{i+1}]$; we then find

$$O([x_{i+1}, y_{i+1}]) \subseteq U_O(x_{i+1}, y_{i+1}) \subseteq [x_{i+1}, y_{i+1}],$$

which concludes our proof. □

This lemma inspires a generalised notion of refinement that works on sets instead of intervals and is based on the same intuitions.

Definition 4.10 (*Refinement*). An *O-refinement* of X is a set Y such that one of the following holds:

— $O(X) \subseteq Y \subseteq X$, or
— $Y = X \setminus U$, where U is a set of ungrounded lattice elements and $O(Y) \subseteq Y$.

Following the analogy with A-refinements, we call the first type of refinement *application refinement* and the second type *unfoundedness refinement*. An O-refinement is *strict* if $X \neq Y$.

If Y is an O-refinement of X and O is clear from the context, we often denote this $X \to Y$.

Definition 4.11 (*Well-founded set-induction*). Let β be an ordinal number. A *well-founded set-induction* (or shortly, set-induction) is a sequence $L = X_0 \to X_1 \to \cdots \to X_\beta$ such that the following hold:

— For limit ordinals λ, $X_\lambda = \bigcap(\{X_\alpha \mid \alpha < \lambda\}$.
— For every $i < \beta$, $X_i \to X_{i+1}$ is a strict O-refinement.

Definition 4.12 (*Terminal set*). A set-induction $(X_i)_{i \leq \beta}$ is *terminal* if X_β has no strict O-refinements. We call Y a *terminal set* of O if there is a terminal set-induction $(X_i)_{i \leq \beta}$ with $X_\beta = Y$.

In what follows, we show that terminal sets exist and also, that the terminal set is unique. Thus, even though in general many different set-inductions of an operator might exist, they all converge to the same set. An essential property is that refinements preserve being closed under O. This property is similar to A-contractingness [Denecker and Vennekens 2007].

PROPOSITION 4.13. *Given a set-induction $(X_i)_{i \leq \beta}$, for all $i \leq \beta$, it holds that $O(X_i) \subseteq X_i$.*

PROOF. We prove this by induction. It is true initially since $X_0 = L$.
This property is preserved in limit ordinals λ since

$$\begin{aligned} O\left(\bigcap\{X_\alpha \mid \alpha < \lambda\}\right) &\subseteq \bigcap\{O(X_\alpha) \mid \alpha < \lambda\} \\ &\subseteq \bigcap\{X_\alpha \mid \alpha < \lambda\}. \end{aligned}$$

Let $X \to Y$ be an O-refinement and suppose $O(X) \subseteq X$. We show that $O(Y) \subseteq Y$. If $X \to Y$ is an application refinement, then $O(X) \subseteq Y \subseteq X$. In this case also $O(Y) \subseteq O(X) \subseteq Y$. If $X \to Y$ is an unfoundedness refinement, then the result is trivial. This indeed shows that being closed under application of O is preserved by O-refinements. □

PROPOSITION 4.14. *Every well-founded induction $(X_i)_{i \leq \beta}$ is decreasing, i.e. for $i < j$, it holds that $X_j \subseteq X_i$.*

PROOF. It is trivial that unfoundedness refinements result in a decreased set and that taking limits results in a decreased set. Proposition 4.13 shows that also application refinements result in a decreased set. The result follows by induction. □

PROPOSITION 4.15. *Every operator O has a terminal set.*

PROOF. Follows from Proposition 4.14: we can extend every non-terminal induction with a strict O-refinement. This results in a strictly decreasing sequence of subsets of 2^L, which must eventually terminate since 2^L is a complete lattice. □

PROPOSITION 4.16. *Let Z be a terminal set of O and let $X \to Y$ be an O-refinement. If $Z \subseteq X$, then $Z \subseteq Y$.*

PROOF. Suppose Z is a terminal set of O; we know that $Z = O(Z)$.

If $X \to Y$ is an application-refinement, then $O(X) \subseteq Y$, hence $Z = O(Z) \subseteq O(X) \subseteq Y$.

On the other hand, suppose $Y = X \setminus U$, where U are ungrounded points. Define $Z' = Z \setminus U$. It then holds that $Z' = Z \setminus U \subseteq X \setminus U = Y$. Also, Z' is a refinement of Z, since all elements in $Z' \setminus Z$ are ungrounded and

$$O(Z') \subseteq O(Y) \cap O(Z) \subseteq Y \cap Z = Z'.$$

Since Z is a terminal set, this refinement cannot be strict, hence, $U \cap Z = \emptyset$ and $Z \subseteq Y$. □

THEOREM 4.17. *All set-inductions converge to the same set.*

PROOF. Let $(X_i)_{i \leq \beta}$ and $(Y_i)_{i \leq \alpha}$ be terminal set-inductions. We show that $X_\beta = Y_\alpha$. From Propositions 4.16, it follows using induction that $X_\beta \subseteq Y_i$ for every $i \leq \alpha$. Hence, also $X_\beta \subseteq Y_\alpha$. The exact same argument can also be used to show that $X_\beta \supseteq Y_\alpha$, hence the result follows. □

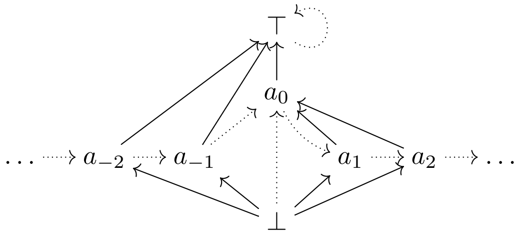
Definition 4.18 (*Well-founded set*). We call the unique set to which all terminal well-founded set-inductions converge the *well-founded set of O* and denote $wfs(O)$.

In AFT, it is known that the A-well-founded fixpoint is more precise than the A-Kripke-Kleene fixpoint for every approximator A. A similar result holds between the Kripke-Kleene set and the well-founded set.

PROPOSITION 4.19. *For every operator O, $wfs(O) \subseteq kks(O)$.*

PROOF. It follows directly from the definition that Kripke-Kleene set-inductions are exactly the well-founded set inductions that only use application refinements. The result then immediately follows from the fact that set-inductions are decreasing (Proposition 4.14). □

Example 4.20. Consider a lattice $\{\bot, \top\} \cup \{a_i \mid i \in \mathbb{Z}\}$ with order as depicted below and an operator O that maps that maps $\bot$ to a_0, $\top$ to $\top$ and every a_i to a_{i+1}.



First, we notice that $\{\bot\} \cup \{a_i \mid i \geq 0\}$ are the O-grounded points. Indeed, $\top$ is ungrounded since $O(\top \wedge a_0) \leq a_0$ and for $i < 0$ it holds that $O(a_i \wedge a_0) = O(\bot) = a_0 \leq a_0$. We also notice that $\top$ is the only fixpoint of O. Hence, O has no grounded fixpoints. We now show that the well-founded set of O is empty. Let $X_0 = L$ and $X_1 = O(L) = L \setminus \{\bot\}$. Then, $O(X_1) = X_1$, hence no application refinements are possible. The observations above guarantee that $U = \{a_i \mid i < 0\} \cup \{\top\}$ is a set of ungrounded points. Let X_2 denote $X_1 \setminus U = \{a_i \mid i \geq 0\}$. Then $O(X_2) \subseteq X_2$ and we can refine X_1 to X_2. We notice that $O(X_2) \neq X_2$ hence more application refinements are possible. Unfoundedness refinement has now removed all ungrounded elements in the $\{a_i \mid i \in \mathbb{Z}\}$. More application refinements will gradually delete every of the remaining elements: $X_i = \{a_j \mid j \geq i-2\}$

for $i > 2$. Hence, this example also illustrates that well-founded inductions can lead to an empty set.

Definition 4.21 (*Total operator*). We call O *total* if its well-founded set is a singleton.

Intuitively, total operators are "good" in the sense that they uniquely determine one lattice point of interest.

Example 4.22 (*Example 4.7 continued*). We saw that the semantic operator of $\mathcal{T} = \{q \Leftrightarrow \neg Kp, r \Leftrightarrow \neg Kq\}$ has a singleton Kripke-Kleene set. Since its unique element $\{\{p, q\}, \{q\}\}$ is a grounded fixpoint of $D_{\mathcal{T}}$, there are no more refinements possible and it follows that $\mathit{wfs}(D_{\mathcal{T}}) = \mathit{kks}(D_{\mathcal{T}})$. Hence $D_{\mathcal{T}}$ is total.

Example 4.23 (*Example 4.8 continued*). With $\mathcal{T} = \{Kp \Rightarrow p\}$, we already know that the Kripke-Kleene set of $D_{\mathcal{T}}$ is $\{\bot, \{\{p\}\}\}$. As $\{\{p\}\}$ is not grounded, a set-induction further refines this set to $\{\bot\}$. This is the well-founded set of $D_{\mathcal{T}}$.

In Section 5, we show that it is not a coincidence that the operators in Examples 4.22 and 4.23 are total; there, we show that the semantic operator of a monotonically stratified theory is always total. First, we summarise how the well-founded set relates to other fixpoints studied in AFT.

THEOREM 4.24. *For every operator O and consistent approximator A of O, let $\mathit{wf}(A)$ denote the A-well-founded fixpoint of O and $\mathit{gf}(O)$ the set of grounded fixpoints of O. It holds that*

$$\mathit{gf}(O) \subseteq \mathit{wfs}(O) \subseteq \mathit{wf}(A).$$

Furthermore, each of these inclusions can be strict.

PROOF. The first inclusion is proven as Proposition 4.26 and the second as Proposition 4.28. In Examples 4.27 and 4.7, respectively, we show that these inclusion can be strict. □

The following corollary clarifies the link with stable fixpoints. As such, it can be applied for instance to the stable semantics for logic programming, or to Reiters extensions of default logic.

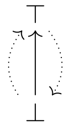
COROLLARY 4.25. *Let A be a consistent approximator of O. All A-stable fixpoints of O are in the well-founded set.*

PROOF. Follows from the fact that all A-stable fixpoints of O are grounded and Theorem 4.24. □

PROPOSITION 4.26. *The well-founded set of O contains all grounded fixpoints of O.*

PROOF. Grounded fixpoints belong to X_0 and are preserved in limit ordinals. As for refinements $X \to Y$, all fixpoints are preserved by applying O, and all grounded points are trivially preserved by unfoundedness refinements. □

Example 4.27. Let L be the lattice $\{\bot, \top\}$ and O the operator that maps $\bot$ to $\top$ and $\top$ to $\bot$.



O has no fixpoints, hence also no grounded fixpoints. There are clearly no strict application refinements of L. Furthermore, since $\bot$ is grounded, there are no strict unfoundedness refinements either. Hence, it holds that $L = wfs(O)$. We notice that the well-founded set can be strictly larger than the set of grounded fixpoints of O.

PROPOSITION 4.28. *The ultimate well-founded fixpoint of an operator O approximates the well-founded set of O.*

PROOF. This follows immediately from Lemma 4.9, which shows that U_O-refinements are also O-refinements. □

COROLLARY 4.29. *Let A be a consistent approximator of O, then the A-well-founded fixpoint of O approximates the well-founded set of O.*

5. LOCALLY MONOTONE OPERATORS

The concept of *stratification* is an important one, in logic programming as well as in autoepistemic and default logic. For a stratified program or theory in these logics, a canonical model can be computed by a procedure of iterated monotone inductions per stratum. This construction is in some sense simpler and perhaps more evident than, e.g., the well-founded model or well-founded set construction. In this section we develop an algebraical notion of stratification for a lattice operator O (such operators are called *locally monotone*) and a corresponding notion of canonical fixpoint. This theory is applicable to a class of operators in arbitrary lattices. We shall prove in Sections 6 and 7 that it generalises the well-known notions of stratification for autoepistemic logic and logic programming respectively. The contribution is two-fold: (1) the notion of locally monotone operator becomes applicable to all applications of fixpoint theory in logic; (2) the algebraical notion of locally monotone operator and its canonical fixpoint provides a sanity check for the well-founded set induction process of the previous section. Indeed, we will prove that all locally monotone operators are *total* and that the well-founded set is the singleton of the canonical fixpoint.

In AEL, a monotonically stratified theory has the property that whenever all symbols from a lower stratum are fixed, the semantic operator on the next stratum is monotone. In logic programming, (locally) stratified programs have a similar property: whenever all atoms up to a certain rank are fixed, the immediate consequence operator is monotone when restricted to atoms of the next rank. Intuitively[6], we generalise these two observations using equivalence relations. We call an operator *locally monotone* if there exists a sequence of equivalence relations $\equiv_i$ on L of increasing precision, such that within every equivalence class of $\equiv_i$, the quotient operator of O modulo $\equiv_{i+1}$ is monotone. For instance, for the case of a (locally) stratified logic programming, the equivalence relations between structures are given by

$$I_1 \equiv_i I_2 \text{ if } I_1 \cap \{p \mid rank(p) \leq i\} = I_2 \cap \{p \mid rank(p) \leq i\},$$

i.e., two structures are equivalent if they agree on all symbols up to rank i. The intended fixpoint of such an operator can then be computed by an iterated least fixpoint computation, each time extending an intended fixpoint of O modulo $\equiv_i$ to an intended fixpoint of O modulo $\equiv_{i+1}$. Before formally introducing locally monotone operators, we introduce some terminology on equivalence relations on lattices.

5.1. Preliminaries: Equivalences, Quotients and Congruences

An equivalence relation on a set is defined as usual. If $\equiv$ is an equivalence relation on a set X, we use $x_\equiv$ for the equivalence class $\{y \mid y \equiv x\}$ and if $Y \subseteq X$, $Y_\equiv$ denotes the set of equivalence classes of elements in Y. Equivalence relations on X naturally extend to the powerset 2^X as follows: for $Y, Z \subseteq X$, $Y \equiv Z$ if $Y_\equiv = Z_\equiv$. We define the quotient

[6] All concepts used in these intuitions are formally defined below.

mapping $p_\equiv : X \to X_\equiv : x \mapsto x_\equiv$. If $x_\equiv$ is an equivalence class, a *representative* of $x_\equiv$ is any $y \in x_\equiv$. If E is a set of equivalence classes, a representative of E is any set Y such that $E = Y_\equiv$.

If $\equiv$ is an equivalence relation on X; relations on X can be pointwise extended to relations on $X_\equiv$. For instance if $\langle L, \leq \rangle$ is a complete lattice, *quotient order of L modulo* $\equiv$, denoted $\leq_\equiv$, is defined as $x_\equiv \leq_\equiv y_\equiv$ if $x \leq y$ for some $x \in x_\equiv, y \in y_\equiv$. Functions f on X (or on 2^X) do not always define a function on the quotient. We call $\equiv$ a *congruence* of f if $f(x) \equiv f(y)$ whenever $x \equiv y$. If $\equiv$ is a congruence of f, the quotient operator $f_\equiv : x_\equiv \mapsto f(x)_\equiv$ is well-defined.

When $\equiv$ is clear from the context, we often abuse notation and omit $\equiv$, i.e., we use $\leq$ for the quotient order on $L_\equiv$. The quotient of a lattice is not always a lattice itself. The following condition is sufficient to ensure this, as shown in Appendix B.

Definition 5.1 (*Meet equivalence*). Let L be a complete lattice and $\equiv$ an equivalence relation on L. We call $\equiv$ a *meet equivalence* if $\equiv$ is a congruence of $\bigwedge : 2^L \to L$.

In other words, $\equiv$ is a meet equivalence if it respects $\wedge$, i.e., if for every two subsets X and Y of L with $X_\equiv = Y_\equiv$, also $\bigwedge X \equiv \bigwedge Y$. If $\equiv$ is a meet equivalence on L, then for every equivalence class $x_\equiv$, it holds that $\{x\} \equiv x_\equiv$, hence also $x = \bigwedge\{x\} \equiv \bigwedge x_\equiv$. It following that $x_\equiv$ has a least element $\bot_{x_\equiv} = \bigwedge x_\equiv$.

PROPOSITION 5.2. *Let $\langle L, \leq \rangle$ be a complete lattice and $\equiv$ a meet equivalence on L. Then $\langle L_\equiv, \leq_\equiv \rangle$ is a complete lattice, called the* quotient lattice of L modulo $\equiv$.

Let $X \subseteq L_\equiv$ be a set of equivalence classes and $X' \subseteq L$ be the set of least representatives of X, (i.e., $X' = \{\bot_x \mid x \in X\}$). Then $(\bigvee X')_\equiv$ is the least upper bound of X in $L_\equiv$ and $(\bigwedge X')_\equiv$ is the greatest lower bound of X in $L_\equiv$.

5.2. Iteration Equivalences

Definition 5.3 (*Iteration equivalence*). Let L be a complete lattice. An *iteration equivalence* on L is an equivalence relation $\equiv$ on L such that

(1) $\equiv$ is a meet equivalence,
(2) for each $x_\equiv \in L_\equiv$, each non-empty chain $C \subseteq x_\equiv$ has a least upper bound in $x_\equiv$, and
(3) whenever $x_\equiv \leq y_\equiv$, there is a $z \in y_\equiv$ with $x \leq z$.

From the fact that an iteration equivalence is a meet equivalence, it follows that
(2') for each $x_\equiv \in L_\equiv$, each non-empty chain $C \subseteq x_\equiv$ has a greatest lower bound in $x_\equiv$, and
(3') whenever $x_\equiv \leq y_\equiv$, there is a $z \in y_\equiv$ such that $x \leq z$.
(For completeness, this is proven as Lemma B.4 and Lemma B.1 respectively).

The above definition is obtained by generalising the situation observed in autoepistemic logic. In AEL, we reasoned in a stratified manner. For instance, in Example 1.1, our first observation was: p does not occur objectively in $\mathcal{T}$, hence in the intended model, it is unknown whether or not p holds. This observation was obtained by reasoning modulo the equivalence relation $Q \equiv Q'$ if $Q|_{\{p\}} = Q'|_{\{p\}}$, i.e., by only reasoning about knowledge of p. After these observations, we focused our attention on the set of possible world structures Q such that $Q \models \neg Kp$ and $Q \models \neg K\neg p$.

Figure 3 depicts the lattice $\mathcal{W}_{\{p,q\}}$ and the equivalence relation that corresponds to "forgetting q", i.e., the relation $\equiv$ such that $Q \equiv Q'$ if and only if $Q|_{\{p\}} = Q'|_{\{p\}}$. First, we notice that the depicted equivalence relation in AEL is a meet equivalence but not a join equivalence (it is not a congruence of $\bigvee$: for instance, $\{\emptyset, \{p\}\} \equiv \{\emptyset, \{p, q\}\}$ but $\{\emptyset, \{p\}\} \vee \{\emptyset, \{p, q\}\} = \{\emptyset\} \not\equiv \{\emptyset, \{p\}\}$). Second, simply being a meet equivalence is insufficient to generalise monotonically stratified theories. The definition of perfect model, in order to compute the least fixpoint, requires us to be able to take limits (least upper bounds) of non-empty chains *within* equivalence classes (for computing the least fixpoint of D_i). Third, we observe that in AEL, the equivalence classes are nicely structured in the sense that whenever $x_\equiv \leq y_\equiv$, there is a $z \in y_\equiv$ such that $x \leq z$.

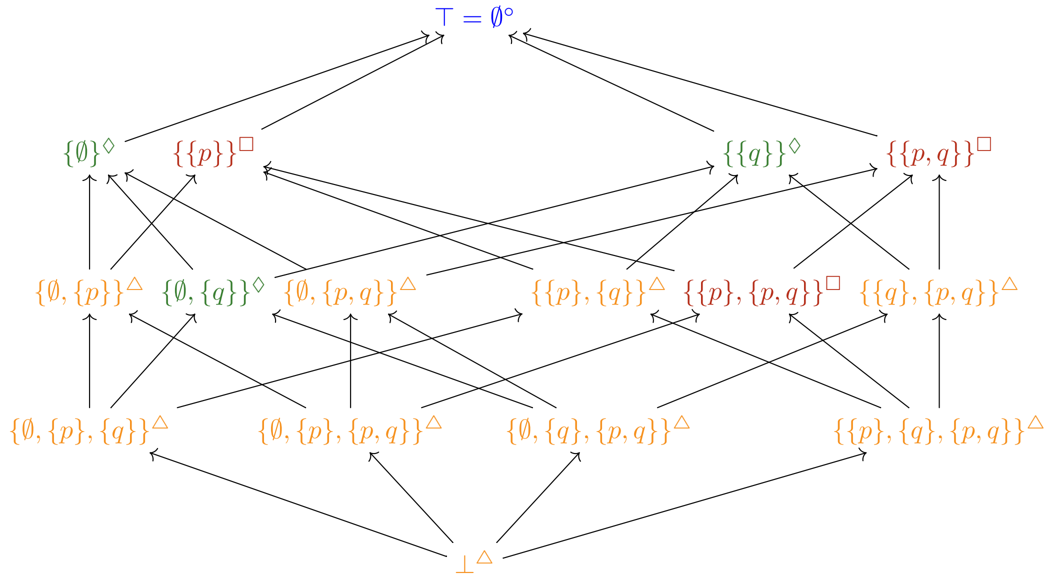

$\top = \emptyset^\circ$
$\{\emptyset\}^\Diamond$
$\{\{p\}\}^\Box$
$\{\{q\}\}^\Diamond$
$\{\{p,q\}\}^\Box$
$\{\emptyset,\{p\}\}^\triangle$
$\{\emptyset,\{q\}\}^\Diamond$
$\{\emptyset,\{p,q\}\}^\triangle$
$\{\{p\},\{q\}\}^\triangle$
$\{\{p\},\{p,q\}\}^\Box$
$\{\{q\},\{p,q\}\}^\triangle$
$\{\emptyset,\{p\},\{q\}\}^\triangle$
$\{\emptyset,\{p\},\{p,q\}\}^\triangle$
$\{\emptyset,\{q\},\{p,q\}\}^\triangle$
$\{\{p\},\{q\},\{p,q\}\}^\triangle$
$\bot^\triangle$

Fig. 3. The lattice $\mathcal{W}_\Sigma$ for $\Sigma = \{p, q\}$. The colours and symbols represent the equivalence classes of lattice elements where two elements are equivalent if they have the same knowledge about p and $\neg p$. The orange elements (annotated with $\triangle$) are those in which both $\neg Kp$ and $\neg K\neg p$ hold. The green ($\Diamond$) elements satisfy $K\neg p$ and $\neg Kp$, the red ($\Box$) satisfy Kp and $\neg K\neg p$ and the blue ($\circ$) Kp and $K\neg p$.

We can order equivalence relations according to how fine-grained they are: one equivalence relation is finer than another if its equivalence classes are smaller; formally $\equiv_2$ is *finer* than $\equiv_1$ (denoted $\equiv_1 \leq_f \equiv_2$) if for all $x, y \in L$ with $x \equiv_2 y$, also $x \equiv_1 y$.

PROPOSITION 5.4. *If $\equiv_1$ and $\equiv_2$ are iteration equivalences of L with $\equiv_1 \leq_f \equiv_2$, then $\equiv_1$ defines an equivalence relation on the lattice $L_{\equiv_2}$. This is an iteration equivalence of $L_{\equiv_2}$.*

5.3. Equivalences and Operators

Suppose $\equiv$ is a congruence of O. In this section, we study how properties of an operator $O_\equiv$ can be used to discover properties of the operator O. For instance: is it possible to find the grounded fixpoints of O from the grounded fixpoints of $O_\equiv$? In general, this will not be possible (e.g., the trivial relation, $x \leq y$ for all $x, y \in L$ is congruence of each operator O). However, again generalising the observations in AEL, we will define algebraical notions that guarantee that this is possible.

First of all, notice that O is not, in general, internal in an equivalence class $x_\equiv$. In fact O is internal in $x_\equiv$ iff $x_\equiv$ is a fixpoint of $O_\equiv$. Suppose this is the case. If $\equiv$ is an iteration equivalence and O is furthermore monotone when restricted to $x_\equiv$, then we can compute a least fixpoint of O inside $x_\equiv$ by standard least fixpoint construction, starting from $\bot_{x_\equiv}$. This is exactly what happens for monotonically stratified theories:

the perfect model of a theory is obtained as the least fixpoint of an operator obtained by "fixing" the value of symbols in lower strata.

This reasoning depends on two properties: *(1)* the fact that $\equiv$ is an iteration equivalence (in order to compute least fixpoints within equivalence classes) and a congruence of O; and *(2)* the fact that O is monotone within equivalence classes. We now formalise these ideas.

Definition 5.5 (*Specialisation*). If $\equiv$ is an iteration equivalence on L, and $\equiv$ is a congruence of O, then we call O a *specialisation* of $O_\equiv$.

Example 5.6. Let $\Sigma = \{p, q\}$ be a vocabulary and $\Sigma_p = \{p\}$ a subvocabulary. Let $\mathcal{P}$ and $\mathcal{P}'$ be the following logic programs

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

$$\mathcal{P}' = \left\{ \; p. \; \right\}.$$

$\mathcal{P}$ and $\mathcal{P}'$ induce immediate consequence operators $T_\mathcal{P}$ and $T_{\mathcal{P}'}$ on the lattice of Σ-structures and Σ_p-structures respectively. The relation $x \equiv y$ if $x|_{\Sigma_p} = y|_{\Sigma_p}$ is an iteration equivalence on 2^Σ. The operator $T_{\mathcal{P}\equiv}$ then equals $T_{\mathcal{P}'}$. Hence, $T_\mathcal{P}$ is a specialisation of $T_{\mathcal{P}'}$, obtained by adding the defining rule of q.

In the following definition, if $O : L \to L$ is an operator and $X \subseteq L$, we use $O|_X$ for the operator $X \to O(X) : x \mapsto O(x)$.

Definition 5.7 ($\equiv$*-monotone operator,* $\equiv$*-constant operator*). Let $\equiv$ be a congruence of O. We call O $\equiv$*-monotone* (respectively $\equiv$*-constant*) if O is monotone (respectively constant) within each equivalence class of $\equiv$, i.e., if $O|_{x_\equiv}$ is monotone (respectively constant) for every equivalence class $x_\equiv \in L_\equiv$.

We will say that O is an $\equiv$-monotone specialisation of $O_\equiv$ if O is $\equiv$-monotone and a specialisation of $O_\equiv$.

Example 5.8 (*Example 5.6 continued*). The operator $T_\mathcal{P}$ is $\equiv$-constant (and hence also $\equiv$-monotone).

Intuitively, for logic programs, suppose $\equiv$ is the equivalence relation that projects out certain variables, i.e., $I \equiv I'$ if $I \cap \Sigma' = I' \cap \Sigma'$. In this case, $T_\mathcal{P}$ is $\equiv$-constant if symbols in $\Sigma \setminus \Sigma'$ do not depend on themselves; $T_\mathcal{P}$ is $\equiv$-monotone if symbols in $\Sigma \setminus \Sigma'$ only depend positively on themselves. In this example, the symbol q which is projected out only depends on p, not on itself.

The following propositions and theorems show that $\equiv$-monotone and $\equiv$-constant specialisations preserve a lot of structure of the operator. Our first result is a study of the relationship between the grounded fixpoints of O and those of $O_\equiv$.

PROPOSITION 5.9. *Suppose O is an $\equiv$-monotone specialisation of $O_\equiv$. If x is grounded for O, then $x_\equiv$ is grounded for $O_\equiv$.*

PROOF. We prove this by contraposition. Choose $x \in L$ such that $x_\equiv$ is not grounded for $O_\equiv$. We show that x is not grounded either. Let $x' = x_\equiv$. Since x' is not grounded, we can choose $v' \in L_\equiv$ such that $O_\equiv(x' \wedge v') \leq v'$ and $x' \not\leq v'$. For every $v \in v'$, it holds that

$$\begin{aligned} p_\equiv(O(x \wedge v)) &= O_\equiv(p_\equiv(x \wedge v)) \\ &= O_\equiv(p_\equiv(x) \wedge p_\equiv(v)) \\ &= O_\equiv(x' \wedge v') \\ &\leq v'. \end{aligned}$$

Hence, using the definition of iteration equivalence, for every $v \in v'$, the set $\{w \in v' \mid w \geq O(x \wedge v)\}$ is non-empty. Since $\equiv$ is a meet equivalence, the greatest lower bound of this set is again an element of v'. Hence, the operator

$$\lambda : v' \to v' : v \mapsto \bigwedge\{w \in v' \mid w \geq O(x \wedge v)\}$$

is well-defined. We know that O is monotone within every equivalence class. Hence, in particular, O is monotone on $x \wedge v'$. Since also $x \wedge \cdot$ and $\bigwedge$ are monotone, we find that λ, the composition of these three is monotone as well. Hence, it is a monotone operator in a chain complete subset of L, thus it has a least fixpoint, which we denote v. Since $\lambda(v) = v$ it holds that

$$v = \bigwedge\{w \in v' \mid w \geq O(x \wedge v)\}.$$

Since $O(x \wedge v) \leq w$ for each $w \in \{w \in v' \mid w \geq O(x \wedge v)\}$, it also holds that

$$O(x \wedge v) \leq \bigwedge\{w \in v' \mid w \geq O(x \wedge v)\} = v.$$

Furthermore $x \not\leq v$ since $p_\equiv(x) = x' \not\leq v' = p_\equiv(v)$. Hence indeed, x is not grounded for O. □

Example 5.10. The converse of Proposition 5.9 does not hold. Consider a lattice $L = \{\bot, \top\}$ with identity operator

$$O : L \to L : x \mapsto x.$$

Since O is monotone, O is a $\equiv$-monotone as well. Furthermore, it is a specialisation of the trivial operator on a singleton lattice (using the equivalence relation $\bot \equiv \top$). However, $\top_\equiv$ is grounded while $\top$ is not.

The converse of Proposition 5.9 does not hold, as the previous example shows. However, a better correspondence holds: Theorem 5.12 shows that there is a one-to-one correspondence between the grounded fixpoints of O and those of $O_\equiv$.

PROPOSITION 5.11. *Let O be an $\equiv$-monotone specialisation of $O_\equiv$. If x' is a grounded fixpoint of $O_\equiv$, then* $\mathrm{lfp}(O|_{x'})$ *is a grounded fixpoint of O.*

PROOF. Suppose $x \overset{\text{def}}{=} \mathrm{lfp}(O|_{x'})$ is not grounded. Then there is a v such that $O(x \wedge v) \leq v$ and $x \not\leq v$. Let v' denote $v_\equiv$. Then also $O_\equiv(x' \wedge v') \leq v'$, and since x' is grounded, $x' \leq v'$. Hence, there exists a w with $v \equiv w$ such that $x \leq w$. Since $x \wedge w = x$ and $v \equiv w$, it holds that $x \wedge v \equiv x$. Now, we find that $O(x \wedge v) \leq O(x) = x$. Since also $O(x \wedge v) \leq v$, $x \wedge v$ is a prefixpoint of the monotone operator $O|_{x'}$ which is smaller than the least fixpoint of that same operator. This yields a contradiction. □

THEOREM 5.12. *Let O be an $\equiv$-monotone specialisation of $O_\equiv$. The mapping $p_\equiv$ induces a one-to-one correspondence between the grounded fixpoints of O and the grounded fixpoints of $O_\equiv$.*

PROOF. Proposition 5.9 guarantees that $p_\equiv$ maps grounded points to grounded points. By definition of $O_\equiv$, it is clear that $p_\equiv$ also maps fixpoints to fixpoints. Now, Proposition 5.11 guarantees that for every grounded fixpoint x' of $O_\equiv$, there is a grounded fixpoint $x = \mathrm{lfp}(O|_{x'})$ of O with $p_\equiv(x) = x'$. Furthermore, since O is monotone within x', all other fixpoints in x' cannot be grounded as they are not minimal. □

The second main result, which we prove below, is that whenever the Kripke-Kleene set of an operator identifies a unique point of interest, then so do ≡-*constant* specialisations. This is not necessarily the case for ≡-*monotone* specialisations.

PROPOSITION 5.13. *Let O be a specialisation of $O_\equiv$. Then $kks(O) \subseteq p_\equiv^{-1}(kks(O_\equiv))$.*

PROOF. We show this by induction. Let

$$X'_0 \to X'_1 \to \cdots \to X'_\beta$$

be a Kripke-Kleene set-induction of $O_\equiv$. We claim that

$$p_\equiv^{-1}(X'_0) \to p_\equiv^{-1}(X'_1) \to \cdots \to p_\equiv^{-1}(X'_\beta)$$

is a Kripke-Kleene-set induction of O. The result easily follows from this claim by taking any *terminal* Kripke-Kleene set-induction.

It is clear that the claim holds for $\beta = 0$ and that this property is preserved in limit ordinals. The fact that it holds for successor ordinals follows from the fact that $O(p^{-1}(X')) \subseteq p^{-1}(O_\equiv(X'))$ for every $X' \subseteq L_\equiv$. □

THEOREM 5.14. *If O is an ≡-constant specialisation of $O_\equiv$ and $kks(O_\equiv)$ is a singleton, then $kks(O)$ is a singleton as well.*

PROOF. Let $kks(O_\equiv) = \{x'\}$ be the Kripke-Kleene set of $O_\equiv$. Proposition 5.13 guarantees that $kks(O) \subseteq p_\equiv^{-1}(\{x'\}) = x'$. Since x' is a fixpoint of $O_\equiv$, $O(x') \subseteq x'$. But, as $O|_{x'}$ is constant, $O(x')$ is a singleton set containing a fixpoint of O, and hence the Kripke-Kleene set of O is indeed a singleton as well. □

Example 5.15. Consider the autoepistemic theory $\mathcal{T} = \{Kp \Rightarrow p\}$ and the equivalence relation $x \equiv y$ for all x, y; the semantic operator $D_\mathcal{T}$ is monotone and hence an ≡-monotone specialisation of the trivial operator on the singleton lattice. However, the Kripke-Kleene set of $D_\mathcal{T}$ is not a singleton. This shows that Theorem 5.14 does not necessarily hold for ≡-*monotone* specialisations.

Our third main result is that ≡-monotone specialisations preserve totality.

PROPOSITION 5.16. *Suppose O is an ≡-monotone specialisation of $O_\equiv$ and $X' \to Y'$ is an $O_\equiv$-refinement, then $p_\equiv^{-1}(X') \to p_\equiv^{-1}(Y')$ is an O-refinement of the same kind.*

PROOF. In this proof, we use X and Y to denote $p_\equiv^{-1}(X')$ and $p_\equiv^{-1}(Y')$ respectively.
If $O_\equiv(X') \subseteq Y' \subseteq X'$, then also

$$p_\equiv^{-1}(O_\equiv(X')) \subseteq p_\equiv^{-1}(Y') = Y \subseteq p_\equiv^{-1}(X') = X.$$

The result then follows from the fact that

$$O(X) = O(p_\equiv^{-1}(X')) \subseteq p_\equiv^{-1}(O_\equiv(X')).$$

On the other hand, suppose $X' \setminus Y'$ consists of ungrounded points and $O_\equiv(Y') \subseteq Y'$. Then by Proposition 5.9 also $X \setminus Y$ consists of ungrounded points. Also, for every $y \in Y$, $O_\equiv(p_\equiv(y)) \in Y'$, hence $O(Y) \subseteq Y$. □

PROPOSITION 5.17. *Let O be an ≡-monotone specialisation of $O_\equiv$. Then $wfs(O) \subseteq p_\equiv^{-1}(wfs(O_\equiv))$.*

PROOF. Let $(X'_i)_{i \leq \beta}$ be a terminal set-induction of $O_{\equiv}$. By iterated application of Proposition 5.16, $(p_{\equiv}^{-1}(X'_i))_{i \leq \beta}$ is a (not necessarily terminal) set-induction of O. Hence, the results follows. □

THEOREM 5.18. *If O is an $\equiv$-monotone specialisation of $O_{\equiv}$ and $O_{\equiv}$ is total, then O is total as well. Furthermore, if $wfs(O_{\equiv}) = \{x'\}$, it holds that $wfs(O) = \{\mathrm{lfp}\,(O|_{x'})\}$*

PROOF. Let x denote $\mathrm{lfp}\,(O|_{x'})$; we claim that $wfs(O) = \{x\}$.

Proposition 5.11 shows that x is a grounded fixpoint of O, thus using Proposition 4.26 we find that $x \in wfs(O)$. Proposition 5.17 guarantees that $wfs(O) \subseteq p_{\equiv}^{-1}(\{x'\}) = x'$. Since O is monotone within x', any well-founded induction that ends in an X with $\{x\} \subseteq X \subseteq x'$ can, by repeated application refinements be reduced to $\{y \in X \mid y \not\leq x\}$. Next, unfoundedness refinement with $v = x$ yields a well-founded induction ending in $\{x\}$. Since $x \in wfs(O)$, no more refinements are possible and this well-founded induction is terminal. Hence $wfs(O) = \{x\}$ as desired. □

The above theory shows how we can construct inteded fixpoints of O from intended fixpoints of $O_{\equiv}$ if O is an $\equiv$-monotone specialisation of $O_{\equiv}$. In the case of AEL, this corresponds to adding one extra stratum to a monotonically stratified theory, where the symbols in the new stratum only depend positively on their own knowledge. It is now clear how to generalise monotonically stratified theories to the algebraical setting, namely as an operator that can be constructed by iteratively applying specialisations starting from the trivial operator on the trivial lattice such that each specialisation is $\equiv_i$-monotone. We call such operators *locally monotone* and use *locally constant* for the stricter definition where at every stage an $\equiv$-constant specialisation is applied (in the context of AEL, we will see that this kind of operators is derived from strict monotonically stratified theories).

Definition 5.19 (*Locally monotone operator, locally constant operator*). Let O be an operator. We call O *locally monotone* (respectively *locally constant*) if there exists a sequence $(\equiv_i)_{i \leq n}$ of iteration equivalences of L such that $\equiv_i \geq_f \equiv_j$ if $i \geq j$ and the following all hold:

— $\equiv_n$ is the identity relation, hence $O_{\equiv_n} = O$,
— $\equiv_0$ is the trivial relation: $\forall x, y \in L : x \equiv_0 y$,
— each $\equiv_i$ is a congruence of O,
— each $O_{\equiv_{i+1}}$ is $\equiv_i$-monotone (respectively $\equiv_i$-constant).

Notice that by Proposition 5.4 $\equiv_{i+1}$ is an iteration equivalence on $L_{\equiv_i}$. Hence it follows that $O_{\equiv_{i+1}}$ is an $\equiv_i$-monotone specialisation of $O_{\equiv_i}$ in the above definition.

The following theorems now follow by repeated application of Theorems 5.14 and 5.18 respectively. They show that locally monotone (and locally constant) operators are “good” in the sense that they determine a unique point of interest. As we will see, for weakly permaconsistent monotonically stratified theories, this unique point of interest is indeed the perfect model.

THEOREM 5.20. *Locally constant operators have a singleton Kripke-Kleene set.*

PROOF. The proof is by induction on the n in Definition 5.19. If O is a locally constant operator and the $\equiv_i$ are as in Definition 5.19, then $O_{\equiv_0}$ trivially has a singleton Kripke-Kleene set. The induction step follows directly from Theorem 5.14. □

THEOREM 5.21. *Locally monotone operators are total.*

PROOF. The proof is by induction on the n in Definition 5.19. If O is a locally monotone operator and the $\equiv_i$ are as in Definition 5.19, then $O_{\equiv_0}$ is trivially total. The induction step follows directly from Theorem 5.18. □

6. LOCALLY MONOTONE OPERATORS IN AUTOEPISTEMIC LOGIC

We now define two semantics for autoepistemic logic of which our algebraic results show that they respect stratification. The first one is based on the concept of a grounded fixpoint, as suggested by Bogaerts et al. [2015]; the second is based on our refinement of the well-founded semantics.

Definition 6.1 (*Grounded fixpoint semantics*). The *grounded fixpoint semantics* for AEL is defined by $Q \models_{gf} \mathcal{T}$ if and only if Q is a grounded fixpoint of $D_{\mathcal{T}}$.

Definition 6.2 (*Well-founded set semantics*). The *well-founded set semantics* for AEL is defined by $Q \models_{wfs} \mathcal{T}$ if and only if $\{Q\} = wfs(D_{\mathcal{T}})$.

Our algebraical results immediately show that the well-founded set semantics is more liberal than the two-valued (ultimate) well-founded semantics which states that Q is a model if and only if (Q, Q) is the (ultimate) well-founded fixpoint: we found that if (Q, Q) is the (ultimate) well-founded fixpoint, then $\{Q\}$ is the well-founded set. We now formalise the relationship between monotonically stratified theories and locally monotone operators, in order to show that the two aforementioned semantics respect stratification.

THEOREM 6.3. *Let $\mathcal{T}$ be a monotonically stratified autoepistemic theory. If $\mathcal{T}$ is weakly permaconsistent, then $D_{\mathcal{T}}$ is locally monotone. Furthermore, if the stratification is strict, then $D_{\mathcal{T}}$ is locally constant.*

PROOF. Suppose $(\mathcal{T}_i)_{0\leq i\leq n}$ is a monotonic stratification of $\mathcal{T}$ with respect to the partition $(\Sigma_i)_{0\leq i\leq n}$. Consider the equivalence relations $\equiv_i$ such that $Q \equiv_i Q'$ if and only if $Q|_{\cup_{j\leq i}\Sigma_j} = Q'|_{\cup_{j\leq i}\Sigma_j}$.

First, it follows directly from the definitions that each of the $\equiv_i$ are iteration equivalences.

Second, each of these equivalences is a congruence of $D_{\mathcal{T}}$: for every interpretation I and possible world structure Q, it holds that $Q, I \models \mathcal{T}$ if and only if $Q, I \models \bigcup_{j\leq i} \mathcal{T}_j$ and $Q, I \models \bigcup_{j>i} \mathcal{T}_j$. For a fixed I, the first is completely determined by the restriction of Q to symbols in strata smaller than (or equal to) i. Furthermore, in $\bigcup_{j>i} \mathcal{T}_j$, the symbols of strata larger than i do not occur objectively, hence the second does not influence $I|_{\cup_{j\leq i}\Sigma_j}$. We conclude that if $Q \equiv_i Q'$, also $D_{\mathcal{T}}(Q) \equiv_i D_{\mathcal{T}}(Q')$: $\equiv_i$ is indeed a congruence of $D_{\mathcal{T}}$.

Third, it is clear that for $i > j$, $\equiv_i \geq_f \equiv_j$.

Fourth, we claim that $(D_{\mathcal{T}})_{\equiv_i} = D_{\cup_{j\leq i}\mathcal{T}_j}$. This follows again from the observation that $Q, I \models \mathcal{T}$ if and only if $Q|_{\cup_{j\leq i}\Sigma_j}, I|_{\cup_{j\leq i}\Sigma_j} \models \bigcup_{j\leq i} \mathcal{T}_j$ and $Q|_{\cup_{j>i}\Sigma_j}, I|_{\cup_{j>i}\Sigma_j} \models \bigcup_{j>i} \mathcal{T}_j$. Since $\mathcal{T}$ is weakly permaconsistent, there is at least one such I for each Q, hence $D_{\mathcal{T}}(Q)$ is non-empty and for every i $D_{\mathcal{T}}(Q)|_{\cup_{j\leq i}\Sigma_j} = D_{\cup_{j\leq i}\mathcal{T}_j}(Q|_{\cup_{j\leq i}\Sigma_j})$, which proves our claim.

Fifth, the fact that each $(D_{\mathcal{T}})_{\equiv_{i+1}}$ is $\equiv_i$-monotone follows immediately from Proposition 3.3.

The five above observations combined indeed yield that $D_{\mathcal{T}}$ is locally monotone; the claim about strict stratifications follows completely analogously. □

Example 6.4. The converse of Theorem 6.3 does not hold. The theory $\{p \Leftrightarrow Kp\}$ is not monotonically stratified; however, its semantic operator is locally monotone and even locally constant. The possible world structure $\{\emptyset\}$, i.e., the world in which we know $\neg p$, is its perfect model, and also its unique model under grounded fixpoint and well-founded set semantics.

COROLLARY 6.5. *Both the grounded fixpoint semantics and the well-founded set semantics respect stratification.*

PROOF. Theorem 6.3 yields that both of these semantics identify a unique fixpoint of interest. The fact that this indeed equals the perfect model of a monotonically stratified theory can be proven inductively. The base case, for monotone operators, is trivial. The induction step is given by Theorem 5.18. □

7. LOCALLY MONOTONE OPERATORS IN LOGIC PROGRAMMING

We now apply our abstract results in the context of logic programming.

As mentioned in the introduction, in this paper, we focus on finite stratifications. In this case, for a locally stratified logic program, we can assign to every atom $p \in \Sigma$ a rank $rank(p) \in \{0, \dots, n\}$ for some $n \in \mathbb{N}$ such that (1) no atom depends on an atom with greater rank, and (2) no atom depends negatively on an atom with the same rank. We now show that $T_\mathcal{P}$ is locally monotone in this case. For every i, we define Σ_i as $\{p \in \Sigma \mid rank(p) \leq i\}$ and $\equiv_i$ such that

$$I \equiv_i I' \text{ if and only if } I \cap \Sigma_i = I' \cap \Sigma_i.$$

We define $\mathcal{P}_i$ as the logic program over Σ_i given by $\mathcal{P}_i = \{r \in \mathcal{P} \mid head(r) \in \Sigma_i\}$. The following proposition reformulates two well-known results about locally stratified logic programs in the lattice terminology used in this paper.

PROPOSITION 7.1. *The following hold.*

- $T_{\mathcal{P}_0}$ *is monotone,*
- *For every* $i \leq n$, $T_{\mathcal{P}_{i+1}}$ *is an* $\equiv_i$*-monotone specialisation of* $T_{\mathcal{P}_i}$.

PROOF. The fact that $T_{\mathcal{P}_0}$ is monotone follows immediately from the fact that all atoms of Σ_0 occur only positively in the body of rules in $T_{\mathcal{P}_0}$.

For every $i \leq n$, the equivalence relation $\equiv_i$ is an iteration equivalence of 2^Σ. It is easy to see that each of the $\equiv_i$ is a congruence of $T_\mathcal{P}$: if two interpretations are identical on Σ_i, since atoms in Σ_i only depend on atoms in Σ_i, applying $T_\mathcal{P}$ will preserve this equivalence.

Furthermore, for each i, it holds that $(T_\mathcal{P})_{\equiv_i} = T_{\mathcal{P}_i}$. The fact that $T_{\mathcal{P}_{i+1}}$ is $\equiv_i$-monotone now follows immediately from the fact that atoms in Σ_{i+1} only depend positively on atoms in Σ_{i+1}. □

THEOREM 7.2. *If* $\mathcal{P}$ *is (locally) stratified, then* $T_\mathcal{P}$ *is locally monotone.*

PROOF. Follows immediately from Proposition 7.1. □

The following result immediately follows from Theorem 7.2, but this result could also have been obtained from the fact that locally stratified logic programs have an exact well-founded model.

COROLLARY 7.3. *If* $\mathcal{P}$ *is (locally) stratified, then* $T_\mathcal{P}$ *is a total operator.*

8. DISCUSSION AND RELATED WORK

Vennekens et al. [2006] have studied modularity of operators. They have results similar to ours, namely that certain fixpoints of interest (stable, well-founded) can be characterised as fixpoints of derived operators on an abstraction of the lattice. There are some key differences between our study and theirs. Firstly, their work focuses on *product lattices*, which are a special case of the equivalence relations we studied here: for product lattice $L = L_1 \times L_2$, the equivalence relation $(x_1, x_2) \equiv (y_1, y_2)$ iff $x_1 = x_2$ is an iteration equivalence. In the context of autoepistemic logic, product lattices do not naturally arise. Vennekens et al. [2006] solved this problem by transforming lattices in AEL to a derived product lattice, but this transformation only preserves semantics for permaconsistent theories. All permaconsistent theories are weakly permaconsistent, but not necessarily the other way round, as witnessed by Example 3.1. Thus, we study a broader class of theories. Secondly, we showed how to construct the unique grounded fixpoint for locally monotone operators stratum per stratum; they showed how well-founded and stable fixpoints can be obtained from the components of stratifiable operators on a product lattice. Our specialisations preserve grounded fixpoints and totality, but do not always preserve stable and well-founded fixpoints; this is good since this work was motivated by an example for which the well-founded semantics fails in the first place. Thirdly, our theory does not require extensions to the bilattice: all concepts are defined in terms of the original operator; no approximations are required.

Our work is closely related to the work by Niemelä [1991]. He also defines a constructive semantics for autoepistemic logic with as main goal that the models (called *L-hierarchic expansions*) are "tightly grounded" in the theory.[7] It deserves to be noted that semantically, his approach works for Example 3.1: the unique L-hierarchic expansion is the intended model in that example. It is currently unknown whether the L-hierarchic expansion semantics respects stratification in general; researching this is a topic for future work. However, the constructive semantics by Niemelä [1991] is not sufficient for our purposes, as the proposed constructions do not follow the reasoning process of a rational introspective agent for several reasons. First of all, in Example 3.1, taking as *enumeration*[8] r, p, q, we would first derive $\neg Kr$, next $\neg Kp$ and finally Kq. I.e., we are able to derive $\neg Kr$ because we will *eventually* derive that q is known. Secondly, there is a lack of *confluence*: a theory such as

$$\{p \Leftrightarrow \neg Kq, q \Leftrightarrow \neg Kp\}$$

can have multiple "constructions" associated to it, resulting in a different final state. If this is the case, these constructions cannot represent the reasoning process of a non-schizophrenic rational agent. An important contribution that distinguishes our work from the previous is that we defined our constructive semantics *algebraically* in AFT. As such, our results are not restricted to Autoepistemic Logic (AEL). For example, locally monotone operators include both the immediate consequence operator of a locally stratified logic program and the semantic operator of a monotonically stratified autoepistemic theory.

9. CONCLUSION

In this paper, we identified a problem with the well-founded semantics for AEL, namely that certain simple and intuitively clear theories have a three-valued well-founded model. We solved this problem by refining the well-founded semantics algebraically. We showed that a large class of lattice operators (called locally monotone operators) have a unique grounded fixpoint, and we provided a constructive characterisation of this fixpoint. This class of operators generalises monotonically stratified autoepistemic theories and (locally) stratified logic programs.

[7] In that work, "grounded" is still an informal concept.

[8] For details and a definition of enumerations, we refer to [Niemelä 1991].

APPENDIX

A. ON INFINITE STRATIFICATIONS

In this appendix, we extend our theory to infinite stratifications. More concretely, we extend the notion of a locally monotone and locally constant operator to allow for a transfinite sequence of equivalence relations. We show that even in this generalised setting, it still holds that all locally constant operators have a singleton Kripke-Kleene set and all locally monotone operators are total.

A.1. Comparing Equivalences

We denote the set of all meet equivalences of L by $\mathbb{E}(L)$ and the set of all iteration equivalences of L by $\mathbb{I}(L)$. We can order equivalence relations according to how fine-grained they are: one equivalence relation is finer than another if its equivalence classes are smaller; we recall the following definition from Section 5.1.

Definition A.1 (*Finer*). Let $\equiv_1$ and $\equiv_2$ be meet equivalences on L. We say that $\equiv_2$ is *finer* than $\equiv_1$ (denoted $\equiv_1 \leq_f \equiv_2$) if for all $x, y \in L$ with $x \equiv_2 y$, also $x \equiv_1 y$.

PROPOSITION A.2. *For every subset $\mathfrak{A}$ of $\mathbb{E}(L)$, the relation $\equiv$ such that $x \equiv y$ if and only if $x \sim y$ for every $\sim$ in $\mathfrak{A}$ is a least upper bound of $\mathfrak{A}$ in $\langle \mathbb{E}(L), \leq_f \rangle$.*

PROOF. It follows directly from the definitions that $\equiv$ is again a meet equivalence. Also, it is clear that $\sim \leq_f \equiv$ for every $\sim$ in $\mathfrak{A}$, and that $\equiv$ is the least relation with this property. □

It follows directly from the previous proposition that $\langle \mathbb{E}(L), \leq_f \rangle$ is chain complete. The finest meet equivalence of L is the identity relation, the least fine is the trivial equivalence relation in which everything is equivalent.

PROPOSITION A.3. *The set $\mathbb{I}(L) \subseteq \mathbb{E}(L)$ is chain complete, i.e., the limit of a (possibly infinite) chain of iteration equivalences is again an iteration equivalence.*

PROOF. Let I be some (totally ordered) index set and $(\equiv_i)_{i \in I}$ a chain of iteration equivalences such that $\equiv_i \leq_f \equiv_j$ if $i \leq j$. Let $\equiv$ be $\bigvee_i \{\equiv_i\}$, we show that $\equiv$ is an iteration equivalence.

First we prove that every equivalence class of $\equiv$ is chain complete. Let $x_\equiv$ be an equivalence class of $\equiv$ and let $C = (c_j)_{j \in J}$ be a non-empty chain in $x_\equiv$. Define $c = \bigvee_j c_j$; we show that $c \in x_\equiv$. First of all, it holds for every i and j that $c_j \in x_{\equiv_i}$, hence C is a chain in $x_{\equiv_i}$. Since $\equiv_i$ is an iteration equivalence, $c \in x_{\equiv_i}$. Thus, using the definition of $\equiv$, it follows that $c \in x_\equiv$.

Secondly, we show that if $x_\equiv \leq y_\equiv$, then there is a $z \in y_\equiv$ such that $x \leq z$. Suppose $x_\equiv \leq y_\equiv$. For every i, it also holds that $x_{\equiv_i} \leq y_{\equiv_i}$, hence we find for every $i \in I$ a non-empty set $Z_i = \{z \in y_{\equiv_i} \mid z \geq x\}$. We define $z_i = \bigwedge Z_i$ for each i. If $i \leq j$, it holds that $\equiv_i \leq_f \equiv_j$, thus $Z_i \supseteq Z_j$ and $z_i \leq z_j$. It follows that $(z_i)_{i \in I}$ is a chain in L and its limit is in each of the $y_{\equiv_i}$. Hence, its limit is indeed an element of $y_\equiv$ greater than x. □

PROPOSITION A.4. *Let $\mathfrak{A}$ be a set of meet equivalences of L. If $x_\equiv \leq y_\equiv$ for every $\equiv$ in $\mathfrak{A}$, then $x_{(\bigvee \mathfrak{A})} \leq y_{(\bigvee \mathfrak{A})}$.*

PROOF. Let $\sim$ denote $\bigvee \mathfrak{A}$. It holds that $\bigvee_{\equiv \in \mathfrak{A}} \bot_{z_\equiv} = \bot_{z_\sim}$ for every $z \in L$. Also, for every $\equiv$ in $\mathfrak{A}$, it holds that $\bot_{x_\equiv} \leq \bot_{y_\equiv}$ since $x_\equiv \leq y_\equiv$. Hence we find that $\bot_{x_\sim} = \bigvee_{\equiv \in \mathfrak{A}} \bot_{x_\equiv} \leq \bigvee_{\equiv \in \mathfrak{A}} \bot_{y_\equiv} = \bot_{y_\sim}$. Thus indeed $x_{(\bigvee \mathfrak{A})} \leq y_{(\bigvee \mathfrak{A})}$. □

A.2. Extended locally monotone operators

We now extend the notion of a locally monotone operator to allow the possibility of an infinite stratification. The definition trivially extends to this setting now that we know how to take limits of chains of iteration equivalences.

Definition A.5 (∞-*locally monotone,* ∞-*locally constant*). Let O be an operator. We call O ∞-*locally monotone* (respectively ∞-*locally constant*) if there exists a sequence $(\equiv_i)_{i\leq\beta}$ of iteration equivalences of L such that $\equiv_i\geq_f\equiv_j$ if $i \geq j$ and the following all hold:

— $\equiv_n$ is the identity relation, hence $O_{\equiv_n} = O$,
— $\equiv_0$ is the trivial relation: $\forall x, y \in L : x \equiv_0 y$,
— each of the $\equiv_i$ is a congruence of O,
— $O_{\equiv_{i+1}}$ is $\equiv_i$-monotone (respectively $\equiv_i$-constant) for every $i < \beta$.
— $\equiv_\lambda = \bigvee(\{\equiv_i \mid i < \lambda\})$ for limit ordinals $\lambda < \beta$.

In order to prove that locally monotone operators are total, most work is already done in Theorem 5.18. The only difficulty left are limit ordinals. We now discuss some properties of limits of equivalences.

PROPOSITION A.6. *Let $(\equiv_i)_{i<\beta}$ be a sequence of iteration equivalences of L that are congruences of O. Let $\equiv$ be $\bigvee(\{\equiv_i \mid i < \beta\})$. Then $x_\equiv$ is a fixpoint of $O_\equiv$ if and only if for every $i < \beta$, $x_{\equiv_i}$ is a fixpoint of $O_{\equiv_i}$.*

PROOF. First suppose $x_{\equiv_i}$ is a fixpoint of $O_{\equiv_i}$ for every i. Hence $O(x) \equiv_i x$, for every i. Now, the definition of $\bigvee(\{\equiv_i \mid i < \beta\})$ guarantees that $O(x) \equiv x$, as desired.
For the other direction, we know that $\equiv\geq_f\equiv_i$ for every i. Hence if $O(x) \equiv x$, also $O(x) \equiv_i x$ for every i. □

PROPOSITION A.7. *Let $(\equiv_i)_{i<\beta}$ be a sequence of iteration equivalences of L that are congruences of O and such that $\equiv_i\geq_f\equiv_j$ whenever $i \geq j$. Let $\equiv$ be $\bigvee(\{\equiv_i \mid i < \beta\})$. If $x_{\equiv_i}$ is grounded for each of the $i < \beta$, then $x_\equiv$ is grounded for $O_\equiv$.*

PROOF.
Suppose that for each of the $i < \beta$, $x_{\equiv_i}$ is grounded for $O_{\equiv_i}$ and that for some v

$$O_\equiv(x_\equiv \wedge v_\equiv) \leq v_\equiv.$$

Proposition 5.4 guarantees that each of the $\equiv_i$ is also an iteration equivalence of $L_\equiv$. Hence certainly $O_{\equiv_i}(x_{\equiv_i} \wedge v_{\equiv_i}) \leq v_{\equiv_i}$ for every $i < \beta$. Since $x_{\equiv_i}$ is grounded for $O_{\equiv_i}$, this means that $x_{\equiv_i} \leq v_{\equiv_i}$ for each i. Hence Proposition A.4 yields that $x_\equiv \leq v_\equiv$. □

PROPOSITION A.8. *Let $(\equiv_i)_{i<\beta}$ be a sequence of iteration equivalences of L that are congruences of O and such that $\equiv_i\geq_f\equiv_j$ whenever $i \geq j$. Let $\equiv$ be $\bigvee(\{\equiv_i \mid i < \beta\})$. If X and Y are sets such that for every i, $X_{\equiv_i} \to Y_{\equiv_i}$ is a $O_{\equiv_i}$-refinement of the first (respectively the second kind), then $X_\equiv \to Y_\equiv$ is a refinement of the first (respectively the second kind).*

PROOF. First of all, we claim that if $A, B \subseteq L$ and $A_{\equiv_i} \subseteq B_{\equiv_i}$ for every i, then $A_\equiv \subseteq B_\equiv$. This claim simply follows from the definition of $\bigvee(\{\equiv_i \mid i < \beta\})$.
If $p_{\equiv_i}(O(X)) = O_{\equiv_i}(X_{\equiv_i}) \subseteq Y_{\equiv_i} \subseteq X_{\equiv_i}$ for every y, then by the previous claim, it also holds that $O_\equiv(X_\equiv) \subseteq Y_\equiv \subseteq X_\equiv$.
On the other hand, suppose $X_{\equiv_i} \setminus Y_{\equiv_i}$ consists of ungrounded points and $O_{\equiv_i}(Y_{\equiv_i}) \subseteq Y_{\equiv_i}$. Then Proposition 5.9 guarantees that also $X \setminus Y$ consists of ungrounded points. Also, for every $y \in Y$ and every i, $O_{\equiv_i}(y_{\equiv_i}) \in Y_{\equiv_i}$, hence by our first claim, $O(Y) \subseteq Y$.
□

PROPOSITION A.9. *Let O be an operator such that the conditions in Proposition A.8 are satisfied, then for every $i < \beta$, $wfs(O_\beta) \subseteq \{x_{\equiv_\beta} \mid x_{\equiv_i} \in wfs(O_{\equiv_i})\}$.*

PROOF. Follows immediately by induction using Propositions A.8 and 5.16. Notice that proposition 5.16 guarantees that the precondition in Proposition A.8 that all refinements are of the same kind is satisfied. □

PROPOSITION A.10. *Let O be an operator such that the conditions in Proposition A.8 are satisfied, then for every $i < \beta$, $kks(O_\beta) \subseteq \{x_{\equiv_\beta} \mid x_{\equiv_i} \in kks(O_{\equiv_i})\}$.*

PROOF. Follows immediately by induction using Proposition A.8. □

THEOREM A.11. *All ∞-locally monotone operators are total.*

PROOF. Let $(\equiv_i)_{i \leq \beta}$ be a sequence of equivalences of L satisfying the conditions in Definition 5.19. We use L_i for $L_{\equiv_i}$ and O_i for $O_{\equiv_i}$.

We prove this by induction on β. For $\beta = 0$, the result is trivial. If the results holds for i, it also holds for $i + 1$ using Theorem 5.18.

Let λ be a limit ordinal and suppose for all $i < \lambda$, O_i is total, we show that O_λ is total. First, we show that $wfs(O_\lambda)$ non-empty. For every $i < \lambda$, let $w_i \in L_i$ denote the unique element in the well-founded set of O_i and let $x_i \in L$ the minimal representative of w_i, i.e., $x_i = \bot_{w_i}$. Since $\equiv_j \geq_f \equiv_i$ for $j \geq i$, it holds that $x_j \in w_i$ if $j \geq i$. Hence, the sequence (x_i) is increasing. Let $x = \bigvee_i x_i$ be the limit of this sequence. Now $x_{\equiv_i} = w_i$, for every i, hence Propositions A.7 and A.6 yield that $x_{\equiv_\lambda}$ is a grounded fixpoint of O_λ. Hence $wfs(O_\lambda)$ is non-empty.

Now, Proposition A.9 yields that:

$$wfs(O_\lambda) \subseteq \bigcap_{i<\lambda} (\{x_{\equiv_\lambda} \mid x_{\equiv_i} = w_i\})$$

Hence, if $x'_{\equiv_\lambda} \in wfs(O_\lambda)$, then for every i, $x'_{\equiv_i} = w_i = x_{\equiv_i}$. Thus, using the definition of $\bigvee(\{\equiv_i \mid i < \lambda\})$, we find that $x_{\equiv_\lambda} = x'_{\equiv_\lambda}$. □

THEOREM A.12. *All ∞-locally constant operators have a singleton Kripke-Kleene set.*

PROOF. The argument is completely analogous to the proof of Theorem A.11. □

B. OMITTED PROOFS

LEMMA B.1. *Let $\equiv$ be a meet equivalence on L and $x', y' \in L_\equiv$. The following are equivalent.*

(1) *$x' \leq_\equiv y'$,*
(2) *there exist $x \in x', y \in y'$ such that $x \leq y$,*
(3) *$\bot_{x'} \leq \bot_{y'}$,*
(4) *for each $y \in y'$, there is an $x \in x'$ with $x \leq y$.*

PROOF.

(1) and (2) are equivalent by definition of the quotient order.

(2) implies (3). If $x \leq y$, it holds that $x = x \wedge y$. Hence

$$\bot_{x'} \equiv x = x \wedge y \equiv \bot_{x'} \wedge \bot_{y'}.$$

Since $\bot_{x'} \equiv \bot_{x'} \wedge \bot_{y'}$ and $\bot_{x'}$ is the least element of x', it must hold that $\bot_{x'} = \bot_{x'} \wedge \bot_{y'}$, i.e., that $\bot_{x'} \leq \bot_{y'}$.

(3) entails (4): we can take

(4) trivially entails (2). □

LEMMA B.2. *Let $\langle L, \leq \rangle$ be a complete lattice and $\equiv$ a meet equivalence on L. Let $X \subseteq L_\equiv$ be a set of equivalence classes and $X' \subseteq L$ be the set of least representatives of X, (i.e., $X' = \{\bot_x \mid x \in X\}$). Then $(\bigwedge X')_\equiv$ is the greatest lower bound of X in $L_\equiv$.*

PROOF. First, we show that $(\bigwedge X')_\equiv$ is a lower bound of X. For every $x \in X$ it holds that $\bot_x \in X'$ hence also that

$$\bot_{(\bigwedge X')_\equiv} \leq \bot_x.$$

Hence, it also holds that $(\bigwedge X')_\equiv \leq_\equiv x$ for every $x \in X$.

Now we show that $(\bigwedge X')_\equiv$ is the greatest lower bound X. If y is another lower bound of X, then it must hold that $\bot_y \leq \bot_x$ for each $x \in X$. Hence, it must hold that $\bot_y \leq \bigwedge\{\bot_x \mid x \in X\}$, i.e., that $y \leq_\equiv (\bigwedge X')_\equiv$. □

LEMMA B.3. *Let $\langle L, \leq \rangle$ be a complete lattice and $\equiv$ a meet equivalence on L. Let $X \subseteq L_\equiv$ be a set of equivalence classes and $X' \subseteq L$ be the set of least representatives of X, (i.e., $X' = \{\bot_x \mid x \in X\}$). Then $(\bigvee X')_\equiv$ is the least upper bound of X in $L_\equiv$.*

PROOF. First, we show that $(\bigvee X')_\equiv$ is an upper bound of X. For every $x \in X$ it holds that

$$\bot_x \leq \bigvee\{\bot_y \mid y \in X\} = \bigvee X',$$

hence, from the definition of $\leq_\equiv$, for every $x \in X$, it holds that

$$x = (\bot_x)_\equiv \leq_\equiv \left(\bigvee X'\right)_\equiv .$$

Now we show that $(\bigvee X')_\equiv$ is the least upper bound of X. If y is another upper bound of X, then it must hold that $\bot_x \leq \bot_y$ for each $x \in X$, hence, it must hold that $\bigvee\{\bot_x \mid x \in X\} \leq \bot_y$. Again, from the definition of $\leq_\equiv$, it follows that $(\bigvee X')_\equiv \leq y$. □

Proposition 5.2. *Let $\langle L, \leq \rangle$ be a complete lattice and $\equiv$ a meet equivalence on L. Then $\langle L_\equiv, \leq_\equiv \rangle$ is a complete lattice, called the* quotient lattice of L modulo $\equiv$.

Let $X \subseteq L_\equiv$ be a set of equivalence classes and $X' \subseteq L$ be the set of least representatives of X, (i.e., $X' = \{\bot_x \mid x \in X\}$). Then $(\bigvee X')_\equiv$ is the least upper bound of X in $L_\equiv$ and $(\bigwedge X')_\equiv$ is the greatest lower bound of X in $L_\equiv$.

PROOF. $L_\equiv$ is clearly a partially ordered set. Lemmas B.2 and B.3 guarantee that $\langle L_\equiv, \leq_\equiv \rangle$ permits least upper bounds and greatest lower bounds. □

It follows easily from the definition of meet equivalence that the following lemma holds.

LEMMA B.4. *If $\equiv$ is a meet equivalence on L, then every chain of representatives of $x_\equiv$ has a (unique) greatest lower bound in $x_\equiv$.*

PROOF. If $X \subseteq x_\equiv$, then $\bigwedge X \equiv \bigwedge\{x\}$ by the definition of meet equivalence. Hence indeed $\bigwedge X \in x_\equiv$. □

Proposition 5.4. *If $\equiv_1$ and $\equiv_2$ are iteration equivalences of L with $\equiv_1 \leq_f \equiv_2$, then $\equiv_1$ defines an equivalence relation on the lattice $L_{\equiv_2}$. This is an iteration equivalence of $L_{\equiv_2}$.*

PROOF. The fact that $\equiv_1$ defines an equivalence relation on $L_{\equiv_2}$ follows from the fact that $\equiv_2$ is finer than $\equiv_1$. The fact that this is a meet equivalence follows directly from the fact that it is a meet equivalence on L. We now show that it is an iteration equivalence.

First, assume that $(c_i)_{i \in I}$ is a chain of elements of $L_{\equiv_2}$ such that for every $i, j \in I$, $c_i \leq_{\equiv_2} c_j$ if $i \leq j$ and $c_i \equiv_1 c_j$.

First, assume that C is a chain of elements of $L_{\equiv_2}$, i.e., a totally ordered subset of $L_{\equiv_2}$ such that for every $c, c' \in C$, it holds that $c \equiv_1 c'$. Since $\equiv_2$ is an iteration equivalence of L, we can choose representatives $d_c in L$ for every $c \in C$ such that $d_c \leq d_{c'}$ if $c \leq c'$. Since all $c \in C$ are equivalent modulo $\equiv_1$, and the d_c are representatives of elements in C, it also holds $d_c \equiv_1 d_{c'}$ if $c, c' \in C$. Thus, $D := \{d_c \mid c \in C\}$ is a chain in L. Let d denote $\bigvee D$. Since $\equiv_1$ is an iteration equivalence, $d \equiv_1 d_c$ for every c. Hence $d_{\equiv_2}$ is the least upper bound of C as desired.

Second, if $x, y \in L_{\equiv_2}$ and $x_{\equiv_1} \leq y_{\equiv_1}$, choose a representative $x' \in L$ of x with respect to $\equiv_2$. Since $\equiv_1 \leq_f \equiv_2$, it also holds that $x'_{\equiv_1} \leq y_{\equiv_1}$. Hence, we find a $z \in L$ such that $z_{\equiv_1} = y_{\equiv_1}$ and such that $x' \leq z$ since $\equiv_1$ is an iteration equivalence on L. Then $x \leq z_{\equiv_2}$, as desired. □

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers, Jan Van den Bussche, Maurice Bruynooghe, Frank Piessens, Thomas Eiter and Gerhard Brewka for their valuable feedback on this work.

REFERENCES

Christian Antic, Thomas Eiter, and Michael Fink. 2013. Hex Semantics via Approximation Fixpoint Theory. In *LPNMR (LNCS)*, Pedro Cabalar and Tran Cao Son (Eds.), Vol. 8148. Springer, 102–115. DOI:http://dx.doi.org/10.1007/978-3-642-40564-8_11

Krzysztof R. Apt, Howard A. Blair, and Adrian Walker. 1988. Towards a Theory of Declarative Knowledge. See Minker [1988], 89–148.

Krzysztof R. Apt and Maarten H. van Emden. 1982. Contributions to the Theory of Logic Programming. *J. ACM* 29, 3 (1982), 841–862. DOI:http://dx.doi.org/10.1145/322326.322339

Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter (Eds.). 2014. *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press. http://www.aaai.org/Library/KR/kr14contents.php

Yi Bi, Jia-Huai You, and Zhiyong Feng. 2014. A Generalization of Approximation Fixpoint Theory and Application. In *Web Reasoning and Rule Systems*, Roman Kontchakov and Marie-Laure Mugnier (Eds.). LNCS, Vol. 8741. Springer International Publishing, 45–59. DOI:http://dx.doi.org/10.1007/978-3-319-11113-1_4

Bart Bogaerts. 2015. *Groundedness in logics with a fixpoint semantics*. Ph.D. Dissertation. Informatics Section, Department of Computer Science, Faculty of Engineering Science. https://lirias.kuleuven.be/handle/123456789/496543 Denecker, Marc (supervisor), Vennekens, Joost and Van den Bussche, Jan (cosupervisors).

Bart Bogaerts, Joost Vennekens, and Marc Denecker. 2015. Grounded fixpoints and their applications in knowledge representation. *Artif. Intell.* 224 (2015), 51–71. DOI:http://dx.doi.org/10.1016/j.artint.2015.03.006

Bart Bogaerts, Joost Vennekens, Marc Denecker, and Jan Van den Bussche. 2014. FO(C): A Knowledge Representation Language of Causality. *TPLP* 14, 4–5-Online-Supplement (2014), 60–69. https://lirias.kuleuven.be/handle/123456789/459436

Gerhard Brewka and Stefan Woltran. 2010. Abstract Dialectical Frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński (Eds.). AAAI Press, 102–111. http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1294

Marc Denecker, Victor Marek, and Mirosław Truszczyński. 2000. Approximations, Stable Operators, Well-Founded Fixpoints and Applications in Nonmonotonic Reasoning. In *Logic-Based Artificial Intelligence*, Jack Minker (Ed.). The Springer International Series in Engineering and Computer Science, Vol. 597. Springer US, 127–144. DOI:http://dx.doi.org/10.1007/978-1-4615-1567-8_6

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

Marc Denecker, Victor Marek, and Mirosław Truszczyński. 2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation* 192, 1 (July 2004), 84–121. DOI:http://dx.doi.org/10.1016/j.ic.2004.02.004

Marc Denecker, Victor Marek, and Mirosław Truszczyński. 2011. Reiter's Default Logic Is a Logic of Autoepistemic Reasoning And a Good One, Too. In *Nonmonotonic Reasoning – Essays Celebrating Its 30th Anniversary*, Gerd Brewka, Victor Marek, and Mirosław Truszczyński (Eds.). College Publications, 111–144. http://arxiv.org/abs/1108.3278

Marc Denecker and Joost Vennekens. 2007. Well-Founded Semantics and the Algebraic Theory of Non-monotone Inductive Definitions. In *LPNMR (Lecture Notes in Computer Science)*, Chitta Baral, Gerhard Brewka, and John S. Schlipf (Eds.), Vol. 4483. Springer, 84–96. DOI:http://dx.doi.org/10.1007/978-3-540-72200-7_9

Marc Denecker and Joost Vennekens. 2014. The Well-Founded Semantics Is the Principle of Inductive Definition, Revisited, See Baral et al. [2014], 1–10. http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7957

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

Melvin Fitting. 1985. A Kripke-Kleene semantics for logic programs. *The Journal of Logic Programming* 2, 4 (1985), 295 – 312. DOI:http://dx.doi.org/10.1016/S0743-1066(85)80005-4

Melvin Fitting. 1986. Partial Models and Logic Programming. *Theor. Comput. Sci.* 48, 3 (1986), 229–255. DOI:http://dx.doi.org/10.1016/0304-3975(86)90096-4

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

Michael Gelfond and Vladimir Lifschitz. 1988. The Stable Model Semantics for Logic Programming. In *ICLP/SLP*, Robert A. Kowalski and Kenneth A. Bowen (Eds.). MIT Press, 1070–1080. http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.6050

S. C. Kleene. 1938. On Notation for Ordinal Numbers. *The Journal of Symbolic Logic* 3, 4 (1938), 150–155. http://www.jstor.org/stable/2267778

Kurt Konolige. 1988. On the relation between default and autoepistemic logic. *Artif. Intell.* 35, 3 (1988), 343–382. DOI:http://dx.doi.org/10.1016/0004-3702(88)90021-5

Hector J. Levesque. 1990. All I Know: A Study in Autoepistemic Logic. *Artif. Intell.* 42, 2-3 (1990), 263–309. DOI:http://dx.doi.org/10.1016/0004-3702(90)90056-6

Victor Marek and Mirosław Truszczyński. 1991. Autoepistemic Logic. *J. ACM* 38, 3 (1991), 588–619. http://dx.doi.org/10.1145/116825.116836

George Markowsky. 1976. Chain-complete posets and directed sets with applications. *algebra universalis* 6, 1 (1976), 53–68. DOI:http://dx.doi.org/10.1007/BF02485815

Jack Minker (Ed.). 1988. *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann.

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

Ilkka Niemelä. 1991. Constructive Tightly Grounded Autoepistemic Reasoning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, John Mylopoulos and Raymond Reiter (Eds.). Morgan Kaufmann, 399–405.

Teodor C. Przymusinski. 1988. On the Declarative Semantics of Deductive Databases and Logic Programs. See Minker [1988], 193–216.

Raymond Reiter. 1980. A Logic for Default Reasoning. *Artif. Intell.* 13, 1-2 (1980), 81–132. DOI:http://dx.doi.org/10.1016/0004-3702(80)90014-4

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

Hannes Strass and Johannes Peter Wallner. 2014. Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory, See Baral et al. [2014], 101–110. http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7917

Maarten H. van Emden and Robert A. Kowalski. 1976. The Semantics of Predicate Logic as a Programming Language. *J. ACM* 23, 4 (1976), 733–742. DOI:http://dx.doi.org/10.1145/321978.321991

Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. 1991. The Well-Founded Semantics for General Logic Programs. *J. ACM* 38, 3 (1991), 620–650. DOI:http://dx.doi.org/10.1145/116825.116838

Joost Vennekens, David Gilis, and Marc Denecker. 2006. Splitting an operator: Algebraic modularity results for logics with fixpoint semantics. *ACM Trans. Comput. Log.* 7, 4 (2006), 765–797. DOI:http://dx.doi.org/10.1145/1182613.1189735

Joost Vennekens, David Gilis, and Marc Denecker. 2007a. Erratum to splitting an operator: Algebraic modularity results for logics with fixpoint semantics (vol 7, pg 765, 2006). (Jan. 2007). DOI:http://dx.doi.org/10.1145/1182613.1189735

Joost Vennekens, Maarten Mariën, Johan Wittocx, and Marc Denecker. 2007b. Predicate Introduction for Logics With a Fixpoint Semantics. Part I: Logic Programming. *Fundamenta Informaticae* 79, 1-2 (September 2007), 187–208. https://lirias.kuleuven.be/handle/123456789/266021

Joost Vennekens, Maarten Mariën, Johan Wittocx, and Marc Denecker. 2007c. Predicate Introduction for Logics With a Fixpoint Semantics. Part II: Autoepistemic Logic. *Fundamenta Informaticae* 79, 1-2 (September 2007), 209–227. https://lirias.kuleuven.be/handle/123456789/146591

Hanne Vlaeminck, Joost Vennekens, Maurice Bruynooghe, and Marc Denecker. 2012. Ordered Epistemic Logic: Semantics, complexity and applications. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Knowledge Representation and Reasoning, Rome, 10-14 July 2012*, Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith (Eds.). AAAI Press, 369–379. http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4513

Received August 2015; revised April 2016; accepted June 2016