Artificial Intelligence 331 (2024) 104110

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming

Jesse Heyninck [a,b,*], Ofer Arieli [c], Bart Bogaerts [d]

[a] *Open Universiteit, the Netherlands*
[b] *University of Cape Town, South Africa*
[c] *School of Computer Science, Tel-Aviv Academic College, Israel*
[d] *Vrije Universiteit Brussel, Belgium*

ARTICLE INFO

Keywords:
Logic Programming
Answer Set Programming
Approximation Fixpoint Theory
Knowledge Representation

ABSTRACT

Approximation fixpoint theory (AFT) is an abstract and general algebraic framework for studying the semantics of nonmonotonic logics. It provides a unifying study of the semantics of different formalisms for nonmonotonic reasoning, such as logic programming, default logic and autoepistemic logic. In this paper, we extend AFT to dealing with *non-deterministic constructs* that allow to handle indefinite information, represented e.g. by disjunctive formulas. This is done by generalizing the main constructions and corresponding results of AFT to non-deterministic operators, whose ranges are sets of elements rather than single elements. The applicability and usefulness of this generalization is illustrated in the context of disjunctive logic programming.

1. Introduction

Semantics of various formalisms for knowledge representation can often be described by fixpoints of corresponding operators. For example, in many logics, theories of a set of formulas can be seen as fixpoints of the underlying consequence operator [55]. Likewise, in logic programming, default logic or formal argumentation, all the major semantics can be formulated as different types of fixpoints of the same operator [23,53]. Such operators are usually non-monotonic, and so one cannot always be sure whether their fixpoints exist, and how they can be constructed.

In order to deal with this ‘illusive nature’ of the fixpoints, Denecker, Marek and Truszczyński [23] introduced a method for *approximating* each value z of the underlying operator by a pair of elements (x, y). These elements intuitively represent lower and upper bounds on z, and so a corresponding *approximation operator* for the original, non-monotonic operator, is constructed. If the approximation operator that is obtained is precision-monotonic, intuitively meaning that more precise inputs of the operator give rise to more precise outputs, then the approximation operator has fixpoints that can be constructively computed, and which in turn approximate the fixpoints of the approximated operator, if such fixpoints exist.

The usefulness of the algebraic theory that underlies the computation process described above was demonstrated on several knowledge representation formalisms, such as propositional logic programming [21], default logic [24], autoepistemic logic [24], abstract argumentation and abstract dialectical frameworks [53], hybrid MKNF [42], the graph description language SHACL [11], and active integrity constraints [10], each one of which was shown to be an instantiation of this abstract theory of approximation. More precisely, it was shown that various semantics of the formalisms mentioned above correspond to the various fixpoints defined

* Corresponding author.
E-mail address: jesse.heyninck@ou.nl (J. Heyninck).

https://doi.org/10.1016/j.artint.2024.104110
Received 2 December 2022; Received in revised form 29 January 2024; Accepted 2 March 2024

Available online 8 March 2024

in [23]. This means that approximation fixpoint theory (AFT, for short) captures the uniform principles underlying all these non-monotonic formalisms in a purely algebraic way.

Besides its unifying capabilities, AFT also allows for a straightforward definition of semantics for new formalisms. Indeed, one merely has to define an approximation of the operator of interest, and AFT then automatically gives rise to a family of semantics with several desirable properties. This potential of AFT for a straightforward derivation of semantics was demonstrated in e.g. logic programming, where it was used to define semantics for extensions of logic programs [3,18,47], in argumentation theory, where it was used to define semantics for (weighted) abstract dialectical frameworks (ADFs, [9,53]), in autoepistemic logic, where it was used for defining distributed variants that are suitable for studying access control policies [35], and in second-order logic extended with non-monotonic inductive definitions [20].

Another benefit of AFT is that, due to its generality, it has proven useful to develop central concepts, such as strong equivalence [56], groundedness [12], safe inductions [13], and stratification [60] for approximation operators in a purely algebraic way, which then allow us to derive, for all the specific formalisms representable in AFT, results on these concepts as straightforward corollaries.

So far, AFT has mainly been applied to *deterministic* operators, i.e., operators which map single inputs to single outputs. This means that, while AFT is able to characterize semantics for normal logic programs (i.e., programs consisting of rules with single atomic formulas as their head), *disjunctive* logic programs [44] (i.e., programs consisting of rules with disjunctions of atoms in their head) cannot be represented by it. The same holds for the representation of e.g. default logic versus disjunctive default logic [33], abstract argumentation versus set-based abstract argumentation [45] and the generalization of abstract dialectical frameworks to *conditional* abstract dialectical frameworks [40].

Extending AFT to handle disjunctive information is therefore a desirable goal, as the latter has a central role in systems for knowledge representation and reasoning, and since disjunctive reasoning capabilities provide an additional way of expressing uncertainty and indeterminism to many formalisms for non-monotonic reasoning. However, the introduction of disjunctive reasoning often increases the computational complexity of formalisms and thus extends their modeling capabilities [26]. Perhaps due to this additional expressiveness, the integration of non-deterministic reasoning with non-monotonic reasoning (NMR) has often proven non-trivial, as witnessed e.g. by the large body of literature on disjunctive logic programming [43,44]. The implementation of non-deterministic reasoning in NMR yielded the formulation of some (open) problems that are related to the combination of non-monotonic and disjunctive reasoning [5,6,14], or was restricted to limited semantics available for the core formalism, as is the case for e.g. default logic [33].

The goal of this work is to provide an adequate framework for modeling disjunctive reasoning in NMR. We do so by extending AFT to handle *non-deterministic operators*. This idea was first introduced by Pelov and Truszczyński in [48], where some first results on two-valued semantics for disjunctive logic programs were provided. In this paper, we further extend AFT for non-deterministic operators. This, among others, allows a generalization of the results of [48] to the three-valued case. In particular, we define several interesting classes of approximating fixpoints and show their existence, constructability and consistency where it is possible. An application of this theory is demonstrated in the context of disjunctive logic programming. Furthermore, we show that our theory is a conservative generalization of the work in [23] of AFT for deterministic operators, in the sense that all the concepts introduced in this paper coincide with the deterministic counterparts when the operator at hand happens to be deterministic.

The outcome of this work is therefore a comprehensive study of semantics for non-monotonic formalisms incorporating non-determinism. Its application is demonstrated in this paper in the context of disjunctive logic programming. Specifically, the paper contains the following contributions:

1. We define variants of both the *Kripke-Kleene* and the *stable/well-founded* semantics, the *interpretation* or *fixpoint semantics* and *state semantics*. Interpretation semantics consist of single pairs of elements, and thus approximate a single element. State semantics, on the other hand, consist of pairs of sets of elements, intuitively viewed as a convex set, which approximates a set of elements.
2. We show that the Kripke-Kleene state and well-founded state (obtained as the least fixpoint of the stable state operator) exist and are unique (Theorem 2 and Theorem 5).
3. We show that the Kripke-Kleene state approximates any fixpoint of an approximation operator (Theorem 5), whereas the well-founded state approximates any stable fixpoint of an approximation operator (Theorem 5). In more detail, any fixpoint of an approximation operator, respectively stable fixpoint of an approximation operator, is an element of the convex set represented by the Kripke-Kleene state, respectively the well-founded state.
4. We show that when restricting the attention to deterministic operators, the theory reduces to deterministic AFT [23] (Remark 5 and Propositions 3, 9 and 11).
5. We show that, just like in deterministic AFT, stable fixpoints are fixpoints that are minimal with respect to the truth order (Proposition 14).
6. We demonstrate the usefulness of our abstract framework by showing how all the major semantics for disjunctive logic programming can be characterized as fixpoints of an approximation operator for disjunctive logic programming. In more detail, the weakly supported models [15] are characterized as fixpoints of the operator $\mathcal{IC}_{\mathcal{P}}$ (Theorem 1), the stable models can be characterized as the stable fixpoints of $\mathcal{IC}_{\mathcal{P}}$ (Theorem 4) and the well-founded semantics by Alcântara, Damásio and Pereira [2] is strongly related to the well-founded state of $\mathcal{IC}_{\mathcal{P}}$ (Theorem 7).

Relation with previous work on non-deterministic approximation fixpoint theory This work extends and improves our work in [38]. The theory has been simplified on several accounts, among others since: (1) we no longer require minimality of the elements of the

Table 1
List of the notations of different types of sets used in this paper.

Elements	Notations	Example	
Elements of $\mathcal{L}$	$\mathcal{L}$	$x, y, \ldots$	x, y
Sets of elements of $\mathcal{L}$	$\wp(\mathcal{L})$	$X, Y, \ldots$	$\{x_1, y_1, x_2, y_2\}$
Pairs of sets of elements of $\mathcal{L}$	$\wp(\mathcal{L})^2$	$\mathbf{X}, \mathbf{Y}, \ldots$	$(\{x_1, x_2, \ldots\}, \{y_1, y_2, \ldots\})$
Sets of sets of elements of $\mathcal{L}$	$\wp(\wp(\mathcal{L}))$	$\mathcal{X}, \mathcal{Y}, \ldots$	$\{\{x_1, x_2\}, \{x_1\}\}$

Table 2
List of the preorders used in this paper.

Preorder	Type	Definition
Element Orders		
$\leq$	$\mathcal{L}$	primitive
$\leq_i, \leq_t$	$\mathcal{L} \times \mathcal{L}$	bilattice orders (Definition 4)
$\leq_i$	$\mathcal{L}^2 \times \mathcal{L}^2$	$(x_1, y_1) \leq_i (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \geq y_2$
$\leq_t$	$\mathcal{L}^2 \times \mathcal{L}^2$	$(x_1, y_1) \leq_t (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \leq y_2$
Set-based Orders		
$\preceq_L^S$	$\wp(\mathcal{L}) \times \wp(\mathcal{L})$	$X \preceq_L^S Y$ iff for every $y \in Y$ there is an $x \in X$ s.t. $x \leq y$
$\preceq_L^H$	$\wp(\mathcal{L}) \times \wp(\mathcal{L})$	$X \preceq_L^H Y$ iff for every $x \in X$ there is an $y \in Y$ s.t. $x \leq y$
$\preceq_i^A$	$\wp(\mathcal{L})^2 \times \wp(\mathcal{L})^2$	$(X_1, Y_1) \preceq_i^A (X_2, Y_2)$ iff $X_1 \preceq_L^S X_2$ and $Y_2 \preceq_L^H Y_1$

Table 3
List of the operators used in this paper.

Operator	Notation	Type	Definition
Non-deterministic operator	O	$\mathcal{L} \to \wp(\mathcal{L})$	Definition 8
Non-deterministic approximation operator	$\mathcal{O}$	$\mathcal{L}^2 \to \wp(\mathcal{L})^2$	Definition 11
Non-deterministic state approx. operator	$\mathcal{O}'$	$\wp(\mathcal{L})^2 \to \wp(\mathcal{L})^2$	Definition 18
Stable operator	$S(\mathcal{O})$	$\mathcal{L}^2 \to \wp(\mathcal{L})^2$	Definition 20

range of a non-deterministic (approximation) operator, which leads to a significant decrease in the number of lattice-constructions needed, (2) the state operator is now more general and can be defined on the basis of a non-deterministic approximation operator. Furthermore, intuitive explanations and illustrative examples are added throughout the paper. As a by-product, the simplified framework allows us to identify and correct a faulty statement on the $\leq_i$-monotonicity of the approximation operator for disjunctive logic programs, made in [38] (see Remark 7).

Both this paper and the work in [38] were partially inspired by Pelov and Truszczyński's work [48] where, to the best of our knowledge, the idea of a non-deterministic operator was first introduced in approximation fixpoint theory. Pelov and Truszczyński [48] studied only two-valued semantics, whereas we study the full range of semantics for AFT. Furthermore, in [48], Pelov and Truszczyński require minimality of the codomain of non-deterministic operators, which we do not require here.

Outline of this paper The rest of this paper is organized as follows: In Section 2 we recall the necessary background on disjunctive logic programming (Section 2.1) and approximation fixpoint theory (Section 2.2). In Section 3 we introduce non-deterministic operators and their approximation, and show some preliminary results on approximations of non-deterministic operators. In Section 4 we study these non-deterministic approximation operators, showing their consistency and introducing and studying their fixpoints, Kripke-Kleene interpretation and the Kripke-Kleene state semantics. In Section 5 we introduce and study the stable interpretation and state semantics, as well as the well-founded state semantics. Related work is discussed in Section 6, followed by a conclusion in Section 7. Proofs of results on disjunctive logic programming are given in Appendix A, and additional characterization results on other semantics for disjunctive logic programs are given in Appendix B.

2. Background and preliminaries

In this section, we recall the necessary basics of approximation fixpoint theory (AFT) for deterministic operators. We start with a brief survey on disjunctive logic programming (DLP, Section 2.1), which will serve to illustrate concepts and results of the general theory of non-deterministic AFT (Section 2.2).

Remark 1. Before proceeding, a note on notation: As we often have to move from the level of single elements to sets of elements, the paper is, by its nature, notationally heavy. We tried to keep the notational burden as light as possible by staying consistent in our notation of different types of elements. For the readers convenience, we provide already at this stage a summary of the notations of different types of sets (Table 1), the preorders (Table 2) and the operators (Table 3) that are used in this paper.

2.1. Disjunctive logic programming

In what follows we consider a propositional[1] language $\mathfrak{L}$, whose atomic formulas are denoted by p, q, r (possibly indexed), and that contains the propositional constants T (representing truth), F (falsity), U (unknown), and C (contradictory information). The connectives in $\mathfrak{L}$ include negation $\neg$, conjunction $\wedge$, disjunction $\vee$, and implication $\leftarrow$. Formulas are denoted by ϕ,ψ (again, possibly indexed). Logic programs in $\mathfrak{L}$ may be divided to different kinds as follows:

- A (propositional) *disjunctive logic program* $\mathcal{P}$ in $\mathfrak{L}$ (a dlp, for short) is a finite set of rules of the form $\bigvee_{i=1}^{n} p_i \leftarrow \psi$, where $\bigvee_{i=1}^{n} p_i$ (the rule's head) is a non-empty disjunction of atoms, and ψ (the rule's body) is a (propositional) formula.
- A rule is called *normal*, if its body is a conjunction of literals (i.e., atomic formulas or negated atoms), and its head is atomic. A program is *normal* if it consists only of normal rules; It is *positive* if there are no negations in the rules' bodies.
- We call a rule *disjunctively normal* if its body is a conjunction of literals (and its head is a non-empty disjunction of atoms). A program is called *disjunctively normal*, if it consists of disjunctively normal rules.

The set of atoms occurring in a logic program $\mathcal{P}$ is denoted $\mathcal{A}_{\mathcal{P}}$. In what follows, we will often leave the reference to the language $\mathfrak{L}$ of $\mathcal{P}$ implicit.

The primary algebraic structure for giving semantics to logic programs in our setting is the four-valued structure $\mathcal{FOUR}$, shown in Fig. 1. This structure was introduced by Belnap [7,8] and later considered by Fitting [29] and others in the context of logic programming. It is the simplest instance of a *bilattice* (Definition 4).[2]

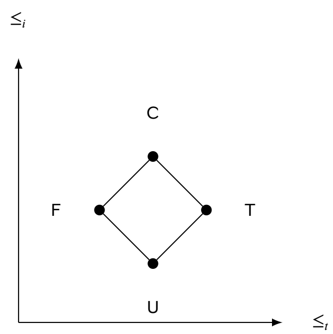


Fig. 1. A four-valued bilattice.

Each element of $\mathcal{FOUR}$ is associated with the propositional constant of $\mathcal{L}$ with the same notation. These elements are arranged in two lattice orders, $\leq_t$ and $\leq_i$, intuitively representing differences in the amount of *truth* and *information* (respectively) that each element exhibits. According to this interpretation, T (respectively, F) exhibits maximal (respectively, minimal) truth, while C (respectively, U) represents "too much" (respectively, "lack of") information. In what follows, we denote by $-$ the $\leq_t$-involution on $\mathcal{FOUR}$ (that is, $-\mathsf{F}=\mathsf{T}$, $-\mathsf{T}=\mathsf{F}$, $-\mathsf{U}=\mathsf{U}$ and $-\mathsf{C}=\mathsf{C}$).

A *four-valued interpretation* of a program $\mathcal{P}$ is a pair (x, y), where $x \subseteq \mathcal{A}_{\mathcal{P}}$ intuitively represents the atoms that are *true* (that, is, strictly true T or contradictory C), and $y \subseteq \mathcal{A}_{\mathcal{P}}$ represents the atoms that are *not false* (i.e., strictly true T or undecided U).[3][4]

Interpretations are compared by two order relations, corresponding to the two partial orders of $\mathcal{FOUR}$:

1. the *information order* $\leq_i$, where $(x, y) \leq_i (w, z)$ iff $x \subseteq w$ and $z \subseteq y$, and
2. the *truth order* $\leq_t$, where $(x, y) \leq_t (w, z)$ iff $x \subseteq w$ and $y \subseteq z$.

The information order represents differences in the "precisions" of the interpretations. Thus, the components of higher values according to this order represent tighter evaluations. The truth order represents increased "positive" evaluations. Truth assignments to complex formulas are then recursively defined as follows:

[1] For simplicity we restrict ourselves to the propositional case. The investigation of disjunctive logic programs with variables is left for future work.

[2] We refer to [30,31] for further details on bilattices and their applications in logic programming.

[3] Somewhat skipping ahead, the intuition here is that x (respectively, y) is a lower (respectively, an upper) approximation of the true atoms.

[4] Notice that we use lower case letters x and y to denote sets, as sets of atoms are elements of the lattice under consideration. This notation is thus consistent with our conventions.

- $(x,y)(p) = \begin{cases} \mathsf{T} & \text{if } p \in x \text{ and } p \in y, \\ \mathsf{U} & \text{if } p \notin x \text{ and } p \in y, \\ \mathsf{F} & \text{if } p \notin x \text{ and } p \notin y, \\ \mathsf{C} & \text{if } p \in x \text{ and } p \notin y. \end{cases}$
- $(x,y)(\neg\phi) = -(x,y)(\phi)$,
- $(x,y)(\psi \wedge \phi) = glb_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$,
- $(x,y)(\psi \vee \phi) = lub_{\leq_t}\{(x,y)(\phi),(x,y)(\psi)\}$.

A four-valued interpretation of the form (x,x) may be associated with a *two-valued* (or *total*) interpretation x, in which for an atom p, $x(p) = \mathsf{T}$ if $p \in x$ and $x(p) = \mathsf{F}$ otherwise. We say that (x,y) is a *three-valued* (or *consistent*) interpretation, if $x \subseteq y$. Note that in consistent interpretations there are no C-assignments.

We now consider semantics for dlp's. First, given a two-valued interpretation, an extension to dlp's of the immediate consequence operator for normal programs [58] is defined as follows:

Definition 1. Given a dlp $\mathcal{P}$ and a two-valued interpretation x, we define:

- $HD_{\mathcal{P}}(x) = \{\Delta \mid \bigvee\Delta \leftarrow \psi \in \mathcal{P} \text{ and } (x,x)(\psi) = \mathsf{T}\}$.
- $IC_{\mathcal{P}}(x) = \{y \subseteq \bigcup HD_{\mathcal{P}}(x) \mid \forall\Delta \in HD_{\mathcal{P}}(x),\ y \cap \Delta \neq \emptyset\}$.

Thus, $IC_{\mathcal{P}}(x)$ consists of sets of atoms, each set contains at least one representative from every disjunction in the head of a rule in $\mathcal{P}$ whose body is x-satisfied (i.e., a representative from each set $\Delta \in HD_{\mathcal{P}}(x)$). In other words, $IC_{\mathcal{P}}(x)$ consists of the two-valued interpretations that validate all disjunctions which are derivable from $\mathcal{P}$ given x. Denoting by $\wp(\mathcal{S})$ the powerset of $\mathcal{S}$, $IC_{\mathcal{P}}$ is a non-deterministic operator on the lattice $\langle \wp(\mathcal{A}_{\mathcal{P}}), \subseteq \rangle$, mapping sets of atoms to sets of sets of atoms.[5]

Example 1. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow\}$. For any two-valued interpretation x, $HD_{\mathcal{P}}(x) = \{\{p,q\}\}$, since $p \vee q \leftarrow \in \mathcal{P}$ and the body of this rule is an empty conjunction and therefore true under any interpretation. Thus, $IC_{\mathcal{P}}(x) = \{\{p\},\{q\},\{p,q\}\}$ for any two-valued interpretation x. This intuitively reflects the fact that p or q has to be true to validate the head of $p \vee q \leftarrow$.

Other semantics for dlp's, this time based on three-valued interpretations, are defined next:

Definition 2. Given a dlp $\mathcal{P}$ and a consistent interpretation (x,y). We say that (x,y) is:

- a *(three–valued) model* of $\mathcal{P}$, if for every $\phi \leftarrow \psi \in \mathcal{P}$, $(x,y)(\phi) \geq_t (x,y)(\psi)$. We denote by $mod(\mathcal{P})$ the set of the three-valued models of $\mathcal{P}$.
- a *weakly supported model* of $\mathcal{P}$, if it is a model of $\mathcal{P}$ and for every $p \in y$, there is a rule $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ such that $p \in \Delta$ and $(x,y)(\phi) \geq_t (x,y)(p)$.

The intuition behind the notions above is the following. An interpretation is a model of $\mathcal{P}$, if for each rule in $\mathcal{P}$ there is at least one atom whose truth value is $\leq_t$-greater or equal to the truth value of the rule's body. Thus, the truth values of the rules' heads in the models of $\mathcal{P}$ are $\leq_t$-greater or equal to the truth values of the rules' bodies. Weakly supported models require that for every atom that is true (respectively undecided), we can find a rule in whose head this atom occurs and for which the body is true (respectively undecided). In other words, every atom is supported by an "activated" rule.

The semantical notions of Definition 2 are illustrated in Example 2 and Example 3 below.

Remark 2. Two-valued weakly supported models are defined by Brass and Dix [15]. Their generalization to the 3-valued case is, to the best of our knowledge, a novel semantics. Brass and Dix [15] also introduce supported semantics, which we characterize in the appendix. An alternative but equivalent definition of a supported model (x,y) is the following: (x,y) is a model, and for every $p \in \mathcal{A}_{\mathcal{P}}$ such that $(x,y)(p) \neq \mathsf{F}$, there is a rule $\bigvee\Delta \leftarrow \phi$ such that $p \in \Delta$ and for every other $p' \in \Delta$, $(x,y)(\phi) \geq_t (x,y)(p) >_t (x,y)(p')$.

Another common way of providing semantics to dlp's is by Gelfond-Lifschitz reduct [32]:

Definition 3. The GL-transformation $\frac{\mathcal{P}}{(x,y)}$ of a disjunctively normal dlp $\mathcal{P}$ with respect to a consistent interpretation (x,y), is the positive program obtained by replacing in every rule in $\mathcal{P}$ of the form

$$p_1 \vee \ldots \vee p_n \leftarrow \bigwedge_{i=1}^{m} q_i \wedge \bigwedge_{j=1}^{n} \neg r_j$$

[5] The operator $IC_{\mathcal{P}}$ is a generalization of the immediate consequence operator from [28, Definition 3.3], where the minimal sets of atoms in $IC_{\mathcal{P}}(x)$ are considered. We will see below that this requirement of minimality is neither necessary nor desirable in the consequence operator.

any negated literal $\neg r_i$ ($1 \leq i \leq k$) by: (1) F if $(x, y)(r_i) = \mathsf{T}$, (2) T if $(x, y)(r_i) = \mathsf{F}$, and (3) U if $(x, y)(r_i) = \mathsf{U}$. In other words, replacing $\neg r_i$ by $(x, y)(\neg r_i)$. To avoid clutter, we denote $\frac{\mathcal{P}}{(x,x)}$ by $\frac{\mathcal{P}}{x}$.

An interpretation (x, y) is a *three-valued stable model* of $\mathcal{P}$ iff it is a $\leq_t$-minimal model of $\frac{\mathcal{P}}{(x,y)}$.[6] For normal logic programs, the *well-founded model* is defined as the $\leq_i$-minimal three-valued stable model, which is unique and guaranteed to exist [49,59].

Example 2. Consider the dlp $\mathcal{P} = \{p \leftarrow \neg p; \quad q \leftarrow \neg r; \quad r \leftarrow \neg q; \quad q \vee r \leftarrow\}$.

- The following interpretations are the (consistent) models of $\mathcal{P}$:

$$(\{p,q,r\},\{p,q,r\}), \quad (\{p,r\},\{p,r\}), \quad (\{q,r\},\{q,r\}), \quad (\{r\},\{p,r\}), \quad (\{q\},\{p,q\}),$$
$$(\{r\},\{p,q,r\}), \quad (\{q\},\{p,q,r\}), \quad (\{p,r\},\{p,q,r\}), \quad (\{p,q\},\{p,q,r\}).$$

  Notice that $(\emptyset, \{p,q,r\})$ is *not* a model of $\mathcal{P}$, since $(\emptyset,\{p,q,r\})(q \vee r) = \mathsf{U} <_t (\emptyset,\{p,q,r\})(\top)$,[7] thus this interpretation is not a model of $q \vee r \leftarrow$.
- The following interpretations are the weakly supported models of $\mathcal{P}$:

$$(\{q\},\{p,q\}), \quad (\{r\},\{p,r\}).$$

- These interpretations are also stable models of $\mathcal{P}$. Indeed, note for instance that:

$$\frac{\mathcal{P}}{(\{q\},\{p,q\})} = \{p \leftarrow \mathsf{U}; \quad q \leftarrow \mathsf{T}; \quad r \leftarrow \mathsf{F}; \quad q \vee r \leftarrow\}.$$

  The minimal (and, in this case also the unique) model of $\frac{\mathcal{P}}{(\{q\},\{p,q\})}$ is $(\{q\},\{p,q\})$ and thus this interpretation is stable.

Example 3. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow q\}$.

- The following interpretations are weakly supported models of $\mathcal{P}$:

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

  One can see that e.g. $(\{p,q\},\{p,q\})$ is *not* weakly supported, as there is no rule for which p is the only atom that is true and occurs in the head (as also q occurs in $p \vee q \leftarrow q$ and is true according to $(\{p,q\},\{p,q\})$).
- The only stable model of $\mathcal{P}$ is $(\emptyset,\emptyset)$. This can be seen by observing that for any interpretation (x, y) it holds that $\frac{\mathcal{P}}{(x,y)} = \mathcal{P}$, and that the minimal model of $\mathcal{P}$ is $(\emptyset,\emptyset)$.

2.2. Approximation fixpoint theory

We first recall some basic algebraic notions. A *lattice* is a partially ordered set (poset) $\langle \mathcal{L}, \leq \rangle$ s.t. for every $x, y \in \mathcal{L}$, a least upper bound $x \sqcup y$ and a greatest lower bound $x \sqcap y$ exist. A lattice is *complete* if every $X \subseteq \mathcal{L}$ has a least upper bound $\bigsqcup X$ and a greatest lower bound $\bigsqcap X$.[8] $\bigsqcup \mathcal{L}$ is denoted by $\top$ and $\bigsqcap \mathcal{L}$ is denoted by $\bot$. A poset X is a *chain* if for all $x, y \in X$, $x \leq y$ or $y \leq x$. It is *chain-complete* if and only if every chain has a least upper bound. A function $f : X \to Y$ from a poset $\langle X, \leq_1 \rangle$ to a poset $\langle Y, \leq_2 \rangle$ is *monotonic* if $x_1 \leq_1 x_2$ implies $f(x_1) \leq_2 f(x_2)$. Given a function $f : X \to X$, we say that $x \in X$ is a *fixpoint* of f if $x = f(x)$, x is called a *pre-fixpoint* of f if $f(x) \leq x$ and x is a *post-fixpoint* of f if $x \leq f(x)$. We define the ordinal powers of a function $f : X \to X$ as follows:

$$f^0(x) = x, \quad f^{\alpha+1}(x) = f(f^\alpha(x)) \text{ for a successor ordinal } \alpha, \quad f^\alpha(x) = \bigsqcup_{\beta<\alpha} f^\beta(x) \text{ for a limit ordinal } \alpha.$$

It follows from Theorem 5.1 shown by Cousot and Cousot [19] that a monotonic operator over a complete lattice admits a least fixpoint that can be constructed by applying the ordinal powers of the operator starting from $\bot$:

Proposition 1 *([19]). Let $\langle \mathcal{L}, \leq \rangle$ be a complete lattice and let $f : \mathcal{L} \to \mathcal{L}$ be a monotonic function. Then there is an ordinal α such that $f^\alpha(\bot)$ is the least fixpoint of f.*

We now recall some basic notions from approximation fixpoint theory (AFT), as described by Denecker, Marek and Truszczyński [23]. As we have already noted, AFT introduces constructive techniques for approximating the fixpoints of an operator O over a lattice $L = \langle \mathcal{L}, \leq \rangle$. This is particularly useful when O is non-monotonic (as is often the case in logic programming,

[6] If $x = y$, (x, y) is called a *two-valued* stable model of $\mathcal{P}$.

[7] We use $\top$ to denote the empty body in the rule $q \vee r \leftarrow$, in order to avoid the potentially confusing $(\emptyset,\{p,q,r\})()$. Notice that $(x, y)(\top) = \top$ for any $x, y \subseteq \mathcal{A}_{\mathcal{P}}$.

[8] As usual, we assume that $\bigsqcup X, \bigsqcap X \in X$.

default logic and abstract argumentation, and other disciplines for non-monotonic reasoning in AI), in which case such operators are not guaranteed to even have a fixpoint, or a unique least fixpoint that can be constructively obtained.

AFT generalizes the principles for the construction of fixpoints to the non-monotonic setting, by working with *approximations* of such operators on a *bilattice* [4,7,8,30,31,34], constructed on the basis of L.

Definition 4. Given a lattice $L = \langle \mathcal{L}, \leq \rangle$, a *bilattice* is the structure $L^2 = \langle \mathcal{L}^2, \leq_i, \leq_t \rangle$, in which $\mathcal{L}^2 = \mathcal{L} \times \mathcal{L}$, and for every $x_1, y_1, x_2, y_2 \in \mathcal{L}$,

- $(x_1, y_1) \leq_i (x_2, y_2)$ if $x_1 \leq x_2$ and $y_1 \geq y_2$,
- $(x_1, y_1) \leq_t (x_2, y_2)$ if $x_1 \leq x_2$ and $y_1 \leq y_2$.[9]

Bilattices of the form L^2 are used for defining operators that approximate operators on L.

An *approximation operator* $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ of an operator $O : \mathcal{L} \to \mathcal{L}$ is an operator that maps every approximation (x, y) of an element z to an approximation (x', y') of the element $O(z)$, thus approximating the behaviour of the approximated operator O. Approximation operators may be viewed as combinations of two operators: $(\mathcal{O}(\cdot,\cdot))_1$ and $(\mathcal{O}(\cdot,\cdot))_2$ which calculate, respectively, a *lower* and an *upper* bounds for the value of O (where, as usual, $(x, y)_1$ respectively $(x, y)_2$ represents the first respectively second component of (x, y)). To avoid clutter, we will also denote $(\mathcal{O}(x, y))_1$ by $\mathcal{O}_l(x, y)$ and $(\mathcal{O}(x, y))_2$ by $\mathcal{O}_u(x, y)$.

Two fundamental requirements on approximation operators are the following:

1. $\leq_i$*-monotonicity:* the values of an approximation operator should be more precise as its arguments are more precise, and
2. *exactness:* exact arguments are mapped to exact values.

These requirements result in the following definition:

Definition 5. Let $O : \mathcal{L} \to \mathcal{L}$ and $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$.

- $\mathcal{O}$ is $\leq_i$*-monotonic*, if when $(x_1, y_1) \leq_i (x_2, y_2)$, also $\mathcal{O}(x_1, y_1) \leq_i \mathcal{O}(x_2, y_2)$; $\mathcal{O}$ is *approximating*, if it is $\leq_i$-monotonic and for any $x \in \mathcal{L}$, $\mathcal{O}_l(x, x) = \mathcal{O}_u(x, x)$.[10]
- $\mathcal{O}$ is an *approximation* of O, if it is $\leq_i$-monotonic and $\mathcal{O}$ *extends* O, that is: $\mathcal{O}(x, x) = (O(x), O(x))$ (for every $x \in \mathcal{L}$).

Notice that any approximation of an operator is an approximation operator.

Remark 3. One can define an approximation operator $\mathcal{O}$ without having to specify which operator O it approximates, and indeed it will often be convenient to study approximation operators without having to refer to the approximated operator. However, one can easily obtain the operator O that $\mathcal{O}$ approximates by letting: $O(x) = \mathcal{O}_l(x, x)$.

Another operator that has a central role in AFT and which is used for expressing the semantics of many non-monotonic formalisms is the *stable operator*, defined next.

Definition 6. For a complete lattice $L = \langle \mathcal{L}, \leq \rangle$, let $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ be an approximation operator. We denote: $\mathcal{O}_l(\cdot, y) = \lambda x.\mathcal{O}_l(x, y)$ and $\mathcal{O}_u(x, \cdot) = \lambda y.\mathcal{O}_u(x, y)$, i.e.: $\mathcal{O}_l(\cdot, y)(x) = \mathcal{O}_l(x, y)$ and $\mathcal{O}_u(x, \cdot)(y) = \mathcal{O}_u(x, y)$. The *stable operator for* $\mathcal{O}$ is: $S(\mathcal{O})(x, y) = (\mathrm{lfp}(\mathcal{O}_l(\cdot, y)), \mathrm{lfp}(\mathcal{O}_u(x, \cdot)))$.

Stable operators capture the idea of minimizing truth, since for any $\leq_i$-monotonic operator $\mathcal{O}$ on $\mathcal{L}^2$, the fixpoints of the stable operator $S(\mathcal{O})$ are $\leq_t$-minimal fixpoints of $\mathcal{O}$ [23, Theorem 4]. Two remarks are of interest here.

1. $\mathcal{O}_l(\cdot, y)$ and $\mathcal{O}_u(x, \cdot)$ are $\leq$-monotonic operators [23, Proposition 20]. This guarantees that the stable operator is well-defined.
2. $S(\mathcal{O})$ is an $\leq_i$-monotonic operator [23, Proposition 20].

Altogether, the following semantic notions are obtained:

Definition 7. Given a complete lattice $L = \langle \mathcal{L}, \leq \rangle$, let $\mathcal{O} : \mathcal{L}^2 \to \mathcal{L}^2$ be an approximation operator. Then:

- (x, y) is a *Kripke-Kleene fixpoint* of $\mathcal{O}$ if it is the $\leq_i$-least (3-valued) stable fixpoint of $\mathcal{O}$.

[9] Recall that we use lower case letters to denote elements of lattice, capital letters to denote sets of elements, and capital calligraphic letters to denote sets of sets of elements (Table 1).

[10] In some papers (e.g. [23]), an approximation operator is defined as a *symmetric* $\leq_i$*-monotonic operator*, i.e., a $\leq_i$-monotonic operator s.t. for every $x, y \in \mathcal{L}$, $\mathcal{O}(x, y) = (\mathcal{O}_l(x, y), \mathcal{O}_l(y, x))$ for some $\mathcal{O}_l : \mathcal{L}^2 \to \mathcal{L}$. However, the weaker condition we take here (taken from [22]) is actually sufficient for most results on AFT.

- (x, y) is a *three-valued stable fixpoint* of $\mathcal{O}$ if $(x, y) = S(\mathcal{O})(x, y)$.
- (x, x) is a *two-valued stable fixpoints* of $\mathcal{O}$ if $(x, x) = S(\mathcal{O})(x, x)$.
- (x, y) is the *well-founded fixpoint* of $\mathcal{O}$ if it is the $\leq_i$-least (3-valued) stable fixpoint of $S(\mathcal{O})$.

Denecker, Marek and Truszczyński [23] show that every approximation operator admits a unique $\leq_i$-minimal fixpoint, and that this is guaranteed to be consistent, i.e. the Kripke-Kleene and well-founded fixpoint are guaranteed to exist, be unique and consistent. Pelov, Denecker and Bruynooghe [47] show that for normal logic programs, the fixpoints based on the four-valued immediate consequence operator for a logic program give rise to the following correspondences: the three-valued stable models coincide with the three-valued semantics as defined by Przymusinski [49], the well-founded model coincides with the homonymous semantics [49,59], and the two-valued stable models coincide with the two-valued (or total) stable models of a logic program.

Example 4. For a normal logic program $\mathcal{P}$, an approximation of the operator $IC_{\mathcal{P}}$ [47] can be obtained by first constructing a lower bound operator as follows[11]:

$$\mathcal{IC}^l_{\mathcal{P}}(x, y) = \{p \in \mathcal{A}_{\mathcal{P}} \mid p \leftarrow \phi \in \mathcal{P}, (x, y)(\phi) \geq_t \mathsf{C}\}$$

and then defining:

$$\mathcal{IC}_{\mathcal{P}}(x, y) = (\mathcal{IC}^l_{\mathcal{P}}(x, y), \mathcal{IC}^l_{\mathcal{P}}(y, x))$$

Notice that the upper bound $\mathcal{IC}^u_{\mathcal{P}}(y, x)$ is defined as $\mathcal{IC}^l_{\mathcal{P}}(y, x)$, i.e. $\mathcal{IC}_{\mathcal{P}}$ is a *symmetric* operator (see also Footnote 10). Pelov, Denecker and Bruynooghe [47] have shown that this operator approximates $IC_{\mathcal{P}}$ for normal logic programs $\mathcal{P}$.

We now illustrate the behaviour of this operator with the following logic program:

$$\mathcal{P} \;=\; \{p \leftarrow \neg q; \quad q \leftarrow \neg p; \quad r \leftarrow r\}.$$

$IC_{\mathcal{P}}$ is thus an operator over the lattice $\langle \wp(\{p, q, r\}), \subseteq\rangle$. Now,

- Concerning the approximation $\mathcal{IC}_{\mathcal{P}}$ of $IC_{\mathcal{P}}$, it holds, e.g. that:
  - $\mathcal{IC}^l_{\mathcal{P}}(\emptyset, \{p, q, r\}) = \emptyset$ as $(\emptyset, \{p, q, r\})(\neg q) = (\emptyset, \{p, q, r\})(\neg p) = (\emptyset, \{p, q, r\})(r) = \mathsf{U}$.
  - $\mathcal{IC}^l_{\mathcal{P}}(\{p, q, r\}, \emptyset) = \{p, q, r\}$ as $(\{p, q, r\}, \emptyset)(\neg q) = (\{p, q, r\}, \emptyset)(\neg p) = (\{p, q, r\}, \emptyset)(r) = \mathsf{C}$.
- We now illustrate the stable operator.
  - It holds that $\mathrm{lfp}(\mathcal{IC}_{\mathcal{P}}(\cdot, \{p\}) = \{p\}$, and therefore $S(\mathcal{IC}^l_{\mathcal{P}})(\{p\}) = \{p\}$.
  - By symmetry of $\mathcal{IC}_{\mathcal{P}}$, $(\{p\}, \{p\})$ is a stable fixpoint of $\mathcal{IC}_{\mathcal{P}}$.
  - Likewise, it can be observed that $(\emptyset, \{p, q\})$ and $(\{q\}, \{q\})$ are stable fixpoints of $\mathcal{IC}_{\mathcal{P}}$.
  - It follows that $(\emptyset, \{p, q\})$ is the well-founded fixpoint of $\mathcal{IC}_{\mathcal{P}}$.

We shall generalize this operator to the disjunctive case in Section 3.

3. Non-deterministic operators and approximations

In this section, we generalize approximation fixpoint theory to allow for non-deterministic operators. In Section 3.1 we formally define non-deterministic operators and the necessary order-theoretic background. In Section 3.2 we then define non-deterministic approximation operators and show some basic results on these operators.

3.1. Non-deterministic operators

In order to characterize (two-valued) semantics for disjunctive logic programming, in [48] Pelov and Truszczyński introduced the notion of non-deterministic operators and accordingly extended AFT to non-deterministic AFT.

Definition 8. A *non-deterministic operator on* $\mathcal{L}$ is a function $O : \mathcal{L} \rightarrow \wp(\mathcal{L}) \setminus \{\emptyset\}$.

Intuitively, a non-deterministic operator assigns to every element x of $\mathcal{L}$ a (nonempty) set of *choices* $O(x) = \{x_1, x_2, \ldots\}$ which can be seen as equally plausible alternatives of the outcome warranted by x. Thus, non-deterministic operators allow for multiple options or choices in their output (reflected by the fact that elements of their codomain are *sets* of elements in the lattice). Just like deterministic operators, it is required that every element x is mapped to at least one choice (which might be the $\leq$-least element $\bot$, if it exists). As an example of non-determinism, consider an activated disjunctive rule $\bigvee \Delta \leftarrow \phi$ (i.e., a rule for which the body is true and its head is a disjunction), requiring that at least one among the disjuncts $\delta \in \Delta$ is true.

[11] Notice that the lattice under consideration is $\langle \wp(\mathcal{A}_{\mathcal{P}}), \subseteq\rangle$, i.e., elements of the lattice are *sets* of atoms.

Example 5. The operator $IC_{\mathcal{P}}$ from Definition 1 is a non-deterministic operator on the lattice $\langle \wp(\mathcal{A}_{\mathcal{P}}), \subseteq \rangle$.

As the ranges of non-deterministic operators are *sets* of lattice elements, one needs a way to compare them. Next, we recall two such relations, known as the *Smyth order* [52] and the *Hoare order* (used in the context of DLP in several works (see, e.g. [2,28])).

Definition 9. Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice, and let $X, Y \in \wp(\mathcal{L})$. Then:

- $X \preceq_L^S Y$ if for every $y \in Y$ there is an $x \in X$ such that $x \leq y$.
- $X \preceq_L^H Y$ if for every $x \in X$ there is a $y \in Y$ such that $x \leq y$.

Here, the sets of lattice elements X and Y represent values of a non-deterministic operator (i.e., these are non-deterministic states). Thus, according to the intuition described above, each one is a set of *choices*. Accordingly, $\preceq_L^S$ means that for every choice y in Y, there is a $\leq$-smaller choice x in X. Likewise, $\preceq_L^H$ means that for every choice x in X, there is a $\leq$-greater choice y in Y. In other words, $\preceq_L^H$ and $\preceq_L^S$ allow to compare non-deterministic states on the basis of the $\leq$-relationship of their constituent elements in L. We will see below that $\preceq_L^S$ is well-suited to compare lower bounds, whereas $\preceq_L^H$ is well-suited to compare upper bounds.

Remark 4. Both $\preceq_L^S$ and $\preceq_L^H$ are preorders (i.e., reflexive and transitive) on $\wp(\mathcal{L})$.

3.2. Non-deterministic approximation operators

We now develop a notion of approximation of non-deterministic operators. Such an approximation generalizes the analogue approximation of deterministic operators explained in Section 2.2. The benefits of such an approximation will be demonstrated in Sections 4 and 5.

Before describing the formal details, we explain the intuition behind the approximation. Recall that, in deterministic AFT, an operator over L is approximated by an *approximation operator* that specifies a lower bound and an upper bound of an approximated element. Thus, an approximation operator essentially consists of two operators over L, the lower bound operator $\mathcal{O}_l$ and the upper bound operator $\mathcal{O}_u$. We generalize this idea to the non-deterministic case. As in deterministic AFT, a non-deterministic approximation operator can be seen as consisting of a non-deterministic lower bound operator and a non-deterministic upper bound operator. A non-deterministic approximation $\mathcal{O}$ of a non-deterministic operator O, maps a pair (x, y) (intuitively representing an approximation of a single value z) to a pair of sets $X, Y \subseteq \mathcal{L}$ (intuitively representing sets of lower bounds X and upper bounds Y on the non-deterministic choices $O(z)$). Thus, an approximation operator $\mathcal{O}$ is of the type $\mathcal{L}^2 \to \wp(\mathcal{L})^2$.

As in the deterministic case, it is natural to assume two formal properties of the approximation operators (recall Definition 5). Below, we adjust the requirements in Definition 5 to the non-deterministic case, using the order relations in Definition 9:

1. *Exactness*: if (x, y) is an approximation of z, every non-deterministic choice $z' \in O(z)$ should have at least one lower bound $x' \in \mathcal{O}_l(x, y)$ and at least one upper bound $y' \in \mathcal{O}_u(x, y)$. In other words, $\mathcal{O}_l(x, y) \preceq_L^S O(z)$ and $O(z) \preceq_L^H \mathcal{O}_u(x, y)$. Informally, every choice in $O(z)$ is in between some lower bound and upper bound of $\mathcal{O}(x, y)$. In the extreme case where the argument is an exact pair (x, x), this means that $\mathcal{O}(x, x) = (O(x), O(x))$ and so, for an exact pair (x, x), both the lower and upper bound operator coincide with the approximated operator O. Thus, $\mathcal{O}(x, x)$ represents a single set of choices. We shall use this as a defining condition: for exact pairs, the lower and upper bound coincide. Intuitively, exact inputs give rise to exact (but non-deterministic) outputs. We will call such an operator *exact*.
2. *Monotonicity*: just like in the deterministic case, a non-deterministic approximation operator should be expected to be monotonic w.r.t. the information ordering: more precise inputs give rise to more precise outputs. To make this notion formally precise, we need a way to compare the precision of pairs of sets (interpreted as a set of lower bounds and a set of upper bounds). A set of lower bounds X_1 is more precise than a set of lower bounds X_2 if, for every lower bound x_1 in the more precise set X_1, there is a less precise (i.e. $\leq$-smaller) lower bound x_2 in the less precise set X_2. Thus: $X_2 \preceq_L^S X_1$. Likewise, a set of upper bounds Y_1 is more precise than a set of upper bounds Y_2 if, for every upper bound y_1 in the more precise set Y_1, there is a less precise (i.e. $\leq$-higher) upper bound y_2 in the less precise set Y_2. Thus: $Y_1 \preceq_L^H Y_2$. Altogether, a pair of bounds (X_1, Y_1) is more precise than a second pair of bounds (X_2, Y_2) if the lower bounds are compared w.r.t. the Smyth-order $\preceq_L^S$ and the upper bounds are compared w.r.t. the Hoare-order $\preceq_L^H$. This idea was defined by Alcântara, Damásio and Moniz Pereira [2] on pairs of sets, and is generalized here to an algebraic setting:

   Definition 10. Given some $X_1, X_2, Y_1, Y_2 \subseteq \mathcal{L}$, we write $(X_1, Y_1) \preceq_i^A (X_2, Y_2)$ iff $X_1 \preceq_L^S X_2$ and $Y_2 \preceq_L^H Y_1$.

We are now ready to define a *non-deterministic approximation operator*: it is an operator $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ assigning to every pair (x, y) a set of lower bounds and a set of upper bounds, that is $\preceq_i^A$-monotonic and for which exact inputs give rise to exact outputs. Formally:

Definition 11. Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice. An operator $\mathcal{O} : \mathcal{L}^2 \to (\wp(\mathcal{L}) \setminus \{\emptyset\})^2$ is called a *non-deterministic approximation operator* (ndao, for short), if satisfies the following properties:

- $\mathcal{O}$ is $\leq_i^A$-monotonic.
- $\mathcal{O}$ is *exact*, i.e., for every $x \in \mathcal{L}$, $\mathcal{O}(x,x) = (\mathcal{O}_l(x,x), \mathcal{O}_l(x,x))$.[12]

We also say that an ndao $\mathcal{O}$ is *an approximation* of the non-deterministic operator, defined as $O(x) = \mathcal{O}_l(x,x)$ (for every $x \in \mathcal{L}$).

In what follows, we sometimes abuse notation and write $(x', y') \in \mathcal{O}(x,y)$ to denote that $x' \in \mathcal{O}_l(x,y)$ and $y' \in \mathcal{O}_u(x,y)$. Likewise, we will often refer to the type of an ndao $\mathcal{O}$ as $\mathcal{L}^2 \to \wp(\mathcal{L})^2$ to avoid clutter (note that this is formally correct and will not cause any confusion).

We remark here that in some works, the definition of an approximation operator assumes a complete lattice. We need this assumption for some, but not all the results. Therefore, we do not make this assumption from the outset, but assume and mention it where this is necessary.

Remark 5. We shall show below (Proposition 3) that Definition 11 extends Definition 5: when an ndao is deterministic, in the sense that $\mathcal{O}(x,y)$ is singleton for every $x, y \in \mathcal{L}$, an ndao reduces to an approximation operator.

Remark 6. It is sometimes useful to assume the following property of ndaos: $\mathcal{O}$ is *symmetric*, if $\mathcal{O}_l(x,y) = \mathcal{O}_u(y,x)$ for any $x, y \subseteq \mathcal{L}$.(Or, equivalently, if $\mathcal{O}_u(x,y) = \mathcal{O}_l(y,x)$ for every $x, y \in \mathcal{L}$). Notice that symmetric operators are exact. Just like in deterministic AFT, the assumption of symmetry is *not* essential and we do not assume it unless specific results require it (see also Footnote 10).

We now give an example of an ndao in the context of disjunctive logic programming. The operator $\mathcal{IC}_\mathcal{P}$ is constructed on the basis of the operators $\mathcal{HD}^l_\mathcal{P}$ and $\mathcal{HD}^u_\mathcal{P}$, which intuitively constitute a lower bound and an upper bound on the activated heads. In more detail, $\mathcal{HD}^l_\mathcal{P}(x,y)$ contains all the heads of rules whose body is at least (according to $\geq_t$) contradictory, i.e., whose body is T or C. Likewise, $\mathcal{HD}^u_\mathcal{P}(x,y)$ contains heads of rules whose body is at least (according to $\geq_t$) undecided, i.e., whose body is T or U. The lower (respectively upper) bounds are then constructed by taking all sets of atoms containing only atoms in at least one of the heads in $\mathcal{HD}^l_\mathcal{P}(x,y)$ (respectively $\mathcal{HD}^u_\mathcal{P}(x,y)$), and containing at least one atom for every head in $\mathcal{HD}^l_\mathcal{P}(x,y)$ (respectively $\mathcal{HD}^u_\mathcal{P}(x,y)$).

Definition 12. For a dlp $\mathcal{P}$ and an interpretation (x,y), we define:

- $\mathcal{HD}^l_\mathcal{P}(x,y) = \{\Delta \mid \bigvee\Delta \leftarrow \phi \in \mathcal{P}, (x,y)(\phi) \geq_t \mathsf{C}\}$,
- $\mathcal{HD}^u_\mathcal{P}(x,y) = \{\Delta \mid \bigvee\Delta \leftarrow \phi \in \mathcal{P}, (x,y)(\phi) \geq_t \mathsf{U}\}$,
- $\mathcal{IC}^l_\mathcal{P}(x,y) = \{x_1 \subseteq \bigcup \mathcal{HD}^l_\mathcal{P}(x,y) \mid \forall \Delta \in \mathcal{HD}^l_\mathcal{P}(x,y),\ x_1 \cap \Delta \neq \emptyset\}$,
- $\mathcal{IC}^u_\mathcal{P}(x,y) = \{y_1 \subseteq \bigcup \mathcal{HD}^u_\mathcal{P}(x,y) \mid \forall \Delta \in \mathcal{HD}^u_\mathcal{P}(x,y),\ y_1 \cap \Delta \neq \emptyset\}$,
- $\mathcal{IC}_\mathcal{P}(x,y) = (\mathcal{IC}^l_\mathcal{P}(x,y), \mathcal{IC}^u_\mathcal{P}(x,y))$.

Example 6. The operator $\mathcal{IC}_\mathcal{P}$ is an approximation of the non-deterministic operator $IC_\mathcal{P}$ in Example 5 (and Definition 1). Furthermore, as we show next, it is a symmetric operator.

Proposition 2. *$\mathcal{IC}_\mathcal{P}$ is a symmetric ndao that approximates $IC_\mathcal{P}$.*

The proof of this proposition and other results on disjunctive logic programs can be found in the appendix.

We now illustrate the operator $\mathcal{IC}_\mathcal{P}$ using two simple disjunctive logic programs:

Example 7. Consider the following dlp: $\mathcal{P} = \{p \vee q \leftarrow \neg q\}$. The corresponding operator $\mathcal{IC}^l_\mathcal{P}$ behaves as follows:

- For any (x,y) with $q \in y$, $\mathcal{HD}^l_\mathcal{P}(x,y) = \emptyset$ and thus $\mathcal{IC}^l_\mathcal{P}(x,y) = \{\emptyset\}$,
- For any (x,y) with $q \notin y$, $\mathcal{HD}^l_\mathcal{P}(x,y) = \{p,q\}$ and thus $\mathcal{IC}^l_\mathcal{P}(x,y) = \{\{p\},\{q\},\{p,q\}\}$.

Since $\mathcal{IC}^l_\mathcal{P}(x,y) = \mathcal{IC}^u_\mathcal{P}(y,x)$ (by Lemma 13), this means that $\mathcal{IC}_\mathcal{P}$ behaves as follows:

- For any (x,y) with $q \notin x$ and $q \notin y$, $\mathcal{IC}_\mathcal{P}(x,y) = (\{\{p\},\{q\},\{p,q\}\},\ \{\{p\},\{q\},\{p,q\}\})$,
- For any (x,y) with $q \notin x$ and $q \in y$, $\mathcal{IC}_\mathcal{P}(x,y) = (\{\emptyset\},\ \{\{p\},\{q\},\{p,q\}\})$,
- For any (x,y) with $q \in x$ and $q \notin y$, $\mathcal{IC}_\mathcal{P}(x,y) = (\{\{p\},\{q\},\{p,q\}\},\ \{\emptyset\})$, and
- For any (x,y) with $q \in x$ and $q \in y$, $\mathcal{IC}_\mathcal{P}(x,y) = (\{\emptyset\},\ \{\emptyset\})$.

Example 8. Consider the dlp $\mathcal{P}$ from Example 2. $\mathcal{IC}^l_\mathcal{P}$ behaves as follows (for arbitrary $y \subseteq \mathcal{A}_\mathcal{P}$):

[12] Recall that we denote by $\mathcal{O}_l$ the operator defined by $\mathcal{O}_l(x,y) = \mathcal{O}(x,y)_1$, and likewise by $\mathcal{O}_u$ the operator defined by $\mathcal{O}_u(x,y) = \mathcal{O}(x,y)_2$.

x	$\mathcal{IC}^l_{\mathcal{P}}(x,y)$
$\emptyset$	$\{\{p,q,r\}\}$
$\{p\}$	$\{\{q,r\}\}$
$\{q\}$	$\{\{p,q\},\{p,q,r\}\}$
$\{r\}$	$\{\{p,r\},\{p,q,r\}\}$
$\{p,q\}$	$\{\{q\},\{q,r\}\}$
$\{p,r\}$	$\{\{r\},\{q,r\}\}$
$\{q,r\}$	$\{\{p,q\},\{p,r\},\{p,q,r\}\}$
$\{p,q,r\}$	$\{\{q\},\{r\},\{q,r\}\}$

Again, by the symmetry of $\mathcal{IC}_{\mathcal{P}}$, the behaviour of $\mathcal{IC}_{\mathcal{P}}$ can be easily derived on the basis of $\mathcal{IC}^l_{\mathcal{P}}$.

In the rest of this section, we discuss several design choices and properties of non-deterministic approximation operators. In particular, firstly, we explain why we do not enforce minimality of the image of an ndao (in contradistinction to other works [3,48]), secondly, we explain how pairs of sets can be interpreted as sets of pairs or convex sets, thirdly, we show a strong relation with non-deterministic operators and, finally, we derive some useful properties of the lower and the upper bound operators.

Remark 7. In the literature (e.g. [3,48]), similar non-deterministic four-valued operators have been defined to characterize the semantics of disjunctive logic programs, inspired by deterministic approximation fixpoint theory. In some of these operators minimality of the image of the operator was built-in. In our setting, this is defined as follows:

- $\mathcal{IC}^{m,l}_{\mathcal{P}}(x,y) = \min_{\subseteq}(\{v \mid \forall \Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y),\ v \cap \Delta \neq \emptyset\})$,[13]
- $\mathcal{IC}^m_{\mathcal{P}}(x,y) = (\mathcal{IC}^{m,l}_{\mathcal{P}}(x,y), \mathcal{IC}^{m,l}_{\mathcal{P}}(y,x))$.

However, there are some issues with this approach, e.g. the operator $\mathcal{IC}^m_{\mathcal{P}}$ is *not* $\preceq^A_i$-monotonic. To see this, consider the program $\mathcal{P} = \{a \vee b \leftarrow;\ \ a \leftarrow c\}$ and the two interpretations $(\emptyset, \{a,b,c\})$ and $(\emptyset, \{a,b\})$. Then we have:

- $\mathcal{IC}^{m,l}_{\mathcal{P}}(\emptyset, \{a,b,c\}) = \{\{a\},\{b\}\}$ and $\mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b,c\}, \emptyset) = \{\{a\}\}$ (the latter since $(\{a,b,c\},\emptyset)(c) = \mathsf{C}$ and thus $\mathcal{HD}^l_{\mathcal{P}}(\{a,b,c\},\emptyset) = \{\{a,b\},\{a\}\}$).
- $\mathcal{IC}^{m,l}_{\mathcal{P}}(\emptyset, \{a,b\}) = \{\{a\},\{b\}\}$ and $\mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b\}, \emptyset) = \{\{a\},\{b\}\}$ (the latter since $(\{a,b,c\},\emptyset)(c) = \mathsf{F}$ and thus $\mathcal{HD}^l_{\mathcal{P}}(\{a,b\},\emptyset) = \{\{a,b\}\}$).

It follows that $\mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b\},\emptyset) \npreceq^H_L \mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b,c\},\emptyset)$ i.e., $\{\{a\},\{b\}\} \npreceq^H_L \{\{a\}\}$, since $\{b\} \nsubseteq \{a\}$. Hence, $\mathcal{IC}^{m,u}_{\mathcal{P}}(\emptyset,\{a,b\}) = \mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b\}, \emptyset) \npreceq^H_L \mathcal{IC}^{m,l}_{\mathcal{P}}(\{a,b,c\},\emptyset) = \mathcal{IC}^{m,u}_{\mathcal{P}}(\emptyset,\{a,b,c\})$. Thus, $(\emptyset,\{a,b,c\}) \leq_i (\emptyset,\{a,b\})$, yet $\mathcal{IC}^m_{\mathcal{P}}(\emptyset,\{a,b,c\}) \npreceq^A_i \mathcal{IC}^m_{\mathcal{P}}(\emptyset,\{a,b\})$, and so $\mathcal{IC}^m_{\mathcal{P}}$ is not $\preceq^A_i$-monotonic.

Note that this is a counter-example to a wrong claim made in [38, Example 2], about $\mathcal{IC}^m_{\mathcal{P}}$ as defined above being a $\preceq^A_i$-monotonic operator.

In the work of Pelov and Truszczyński [48], and of Antić, Eiter and Fink [3], $\preceq^A_i$-monotonicity is not studied, and only $\preceq^S_L$-monotonicity of the lower bound operator is shown and used (i.e., if $(x_1,y_1) \leq_i (x_2,y_2)$ then $\mathcal{IC}^{m,l}(x_1,y_1) \preceq^S_L \mathcal{IC}^{m,l}(x_2,y_2)$). Thus, the above counter-example does not invalidate this claim (i.e., this counter-example does not show that $\mathcal{IC}^{m,l}$ is not $\preceq^S_L$-monotonic). For two-valued stable semantics, this suffices, but as we shall see in what follows, when moving to three- and four-valued semantics, full $\preceq^A_i$-monotonicity is needed.

Altogether, this example shows that requiring minimality in the non-deterministic approximation operator leads to undesirable behaviour. This is perhaps not surprising, as in deterministic approximation theory minimization is also not ensured by the operator itself, but by taking the stable fixpoints of an operator. In our work, minimization is not demanded in the definitions of the operators, but rather is achieved in the definitions of stable operators and fixpoints. And indeed, we will be able to show that this works well, as stable fixpoints will be shown to be $\leq_t$-minimal fixpoints of an ndao (see Proposition 14).

We now explain how pairs of sets can equivalently be interpreted as convex sets or as sets of pairs.

Remark 8. A pair of sets can alternatively be viewed as a *convex set*. A convex set is a set $X \subseteq \mathcal{L}$ that contains no “holes”, i.e., for any $x, y \in X$, if $x \leq z \leq y$ then also $z \in X$. We can then view a pair of sets as a convex set by viewing the two sets as a lower and an upper bound of a convex set. In that case, $\preceq^A_i$ reduces to comparing convex sets in terms of subset relations. This representation will play an important role in what we call the *state semantics* (see Section 4.3).

Remark 9. A third way (in addition to those in the previous remarks) of viewing pairs of sets (which we conceived of as lower and upper bounds) is as set of pairs, i.e., a set of pairs of lower and upper bounds. This is, of course, done by taking all combinations

[13] Recall that $\min_{\subseteq}(X) = \{x \in X \mid \nexists y \in X : y \subset x\}$.

of lower and upper bounds. Also in that case, the order $\preceq_i^A$ makes intuitive sense. In more detail, it boils down to comparing the resulting sets of pairs using $\leq_i$ and the Smyth-ordering.

Definition 13. Given some $\mathbf{X}, \mathbf{Y} \subseteq \mathcal{L}^2$, $\mathbf{X} \preceq_i^S \mathbf{Y}$ if for every $(y_1, y_2) \in \mathbf{Y}$, there is some $(x_1, x_2) \in \mathbf{X}$ s.t. $(x_1, x_2) \leq_i (y_1, y_2)$.

Intuitively, if (x_2, y_2) is more precise than (x_1, y_1), then each interval in $\mathcal{O}(x_2, y_2)$ should be at least as precise as at least one interval in $\mathcal{O}(x_1, y_1)$. In other words, on more precise inputs, the produced intervals become more precise than (some) interval produced on the less precise input. Thus, whereas $\preceq_i^A$ allows for comparison of a set of lower bounds and a set of upper bounds, $\preceq_i^S$ allows for the comparison of two sets of pairs. The order $\preceq_i^A$ over pairs of sets is equivalent to $\preceq_i^S$ over pairs obtained on the basis of a pair of sets:

Lemma 1. *Let some $X_1, X_2, Y_1, Y_2 \subseteq \mathcal{L}$ be given. Then $(X_1, Y_1) \preceq_i^A (X_2, Y_2)$ iff $X_1 \times Y_1 \preceq_i^S X_2 \times Y_2$.*

Proof. [⇒]: Suppose that $X_1 \times Y_1 \preceq_i^A X_2 \times Y_2$ and consider some $(x_2, y_2) \in X_2 \times Y_2$. Since $X_1 \times Y_1 \preceq_i^A X_2 \times Y_2$, there is some $x_1 \in X_1$ s.t. $x_1 \leq x_2$ and there is some $y_1 \in Y_1$ s.t. $y_2 \leq y_1$. Thus, there is an $(x_1, y_1) \in X_1 \times Y_1$ such that $(x_1, y_1) \leq_i (x_2, y_2)$, and so $X_1 \times Y_1 \preceq_i^S X_2 \times Y_2$.

[⇐]: Suppose that $X_1 \times Y_1 \preceq_i^S X_2 \times Y_2$ and consider some $y_2 \in Y_2$. (The case for $x_2 \in X_2$ is analogous.) Then for every $x \in X_2$, $(x, y_2) \in X_2 \times Y_2$ and, since $X_1 \times Y_1 \preceq_i^S X_2 \times Y_2$, there is some $(x_1, y_1) \in X_1 \times Y_1$ s.t. $(x_1, y_1) \leq_i (x, y_2)$, which implies that $y_2 \leq y_1$. Thus, there is some $y_1 \in Y_1$ s.t. $y_2 \leq y_1$, and so $Y_2 \preceq_L^H Y_1$. The proof that $X_1 \preceq_L^S X_2$ is similar, and so $X_1 \times Y_1 \preceq_i^A X_2 \times Y_2$. □

By Lemma 1 it immediately follows that an operator $\mathcal{L}^2 \to \wp(\mathcal{L}) \times \wp(\mathcal{L})$ is $\preceq_i^A$-monotonic if and only if the corresponding operator $\mathcal{L}^2 \to \wp(\mathcal{L}^2)$ (obtained by taking the Cartesian product of the lower and upper bound) is $\preceq_i^S$-monotonic.

Remark 10. The notion of exactness that we introduced constitutes an attempt to generalize the notion of exactness as known from works on non-deterministic AFT, where an interval is exact if the lower and upper bound coincide. We thus generalize this by stating that an operator is exact if, when applied to an exact input, the resulting lower bound and upper bound coincide. Notice that this means that if we interpret a pair of sets as a set of pairs, this means that the set of pairs *includes* exact pairs, but does not necessarily consist *only* of exact pairs. For instance, considering the program $\mathcal{P} = \{p \vee q \leftarrow\}$, $\mathcal{IC}_{\mathcal{P}}(\emptyset, \emptyset) = (\{\{p\}, \{q\}, \{p, q\}\}, \{\{p\}, \{q\}, \{p, q\}\})$, which, interpreted as a set of pairs includes $(\{p\}, \{p\})$, $(\{q\}, \{q\})$ and $(\{p, q\}, \{p, q\})$, but also e.g. $(\{p\}, \{p, q\})$, $(\{q\}, \{p\})$ and $(\{p, q\}, \{q\})$.

Next, we show that when an ndao is deterministic, in the sense that $\mathcal{O}(x, y)$ is a singleton for every $x, y \in \mathcal{L}$, then the ndao reduces to an approximation operator. In other words, our notion of an ndao is a faithful generalization of a deterministic approximation operator.

Proposition 3. *Let an ndao $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ be given s.t. $\mathcal{O}(x, y)$ is a pair of singleton sets for every $x, y \in \mathcal{L}$. Then $\mathcal{O}^{\mathsf{AFT}}$ defined by $\mathcal{O}^{\mathsf{AFT}}(x, y) = (w, z)$ where $\mathcal{O}(x, y) = (\{w\}, \{z\})$, is an approximation operator.*

Proof. We have to show that $\mathcal{O}$ satisfies $\leq_i$-monotonicity and for every $x \in \mathcal{L}$, $\mathcal{O}_l^{\mathsf{AFT}}(x, x) = \mathcal{O}_u^{\mathsf{AFT}}(x, x)$ (according to Definition 5). We first show $\leq_i$-monotonicity. Suppose $(x_1, y_1) \leq_i (x_2, y_2)$. Then by $\preceq_i^A$-monotonicity of $\mathcal{O}$ and Lemma 1, $\mathcal{O}(x_1, y_1) \preceq_i^S \mathcal{O}(x_2, y_2)$. Thus, for every $(w_2, z_2) \in \mathcal{O}(x_2, y_2)$, there is some $(w_1, z_1) \in \mathcal{O}(x_1, y_1)$ such that $(w_1, z_1) \leq_i (w_2, z_2)$. Since both $\mathcal{O}(x_1, y_1)$ and $\mathcal{O}(x_2, y_2)$ are pairs of singleton sets, we obtain $\mathcal{O}(x_1, y_1) \leq_i \mathcal{O}(x_2, y_2)$. For the second condition, notice that $\mathcal{O}_u(x, x) = \mathcal{O}_l(x, x)$ in view of $\mathcal{O}$ being exact. This implies that $\mathcal{O}^{\mathsf{AFT}}(x, x) = (\mathcal{O}_u^{\mathsf{AFT}}(x, x), \mathcal{O}_l^{\mathsf{AFT}}(x, x))$. □

The following lemma shows that an ndao is composed of a $\preceq_L^S$-monotonic lower-bound operator and a $\preceq_L^H$-anti-monotonic upper-bound operator, i.e., for any ndao $\mathcal{O}$, $\mathcal{O}_l$ is $\preceq_L^S$-monotonic in the first argument and $\preceq_L^S$-monotonic in the second argument (and similarly for $\mathcal{O}_u$):

Lemma 2. *An operator $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ is $\preceq_i^A$-monotonic iff for every $x, y \in \mathcal{L}$, $\mathcal{O}_l(\cdot, y)$ is $\preceq_L^S$-monotonic, $\mathcal{O}_l(x, \cdot)$ is $\preceq_L^S$-anti-monotonic, $\mathcal{O}_u(x, \cdot)$ is $\preceq_L^H$-monotonic and $\mathcal{O}_u(\cdot, y)$ is $\preceq_L^H$-anti-monotonic.*

Proof. [⇒]: We first show the $\preceq_L^S$-anti-monotonicity of $\mathcal{O}_l(x, \cdot)$. Consider some $y', y \in \mathcal{L}$ s.t. $y' \leq y$. Then $(x, y) \leq_i (x, y')$ and thus $\mathcal{O}(x, y) \preceq_i^A \mathcal{O}(x, y')$, which in particular means that $\mathcal{O}_l(x, y) \preceq_L^S \mathcal{O}_l(x, y')$.

We now show the case for $\mathcal{O}_u(x, \cdot)$. Consider some $x, y', y \in \mathcal{L}$ s.t. $y' \leq y$. Then $(x, y) \leq_i (x, y')$ and thus $\mathcal{O}(x, y) \preceq_i^A \mathcal{O}(x, y')$. This means that $\mathcal{O}_u(x, y') \preceq_L^H \mathcal{O}_u(x, y)$.

$\preceq_L^S$-monotonicity of $\mathcal{O}_l(\cdot, y)$ and $\preceq_L^H$-monotonicity of $\mathcal{O}_u(\cdot, y)$ are shown similarly.

[⇐]: Suppose that for every $x, y \in \mathcal{L}$, (1) $\mathcal{O}_l(\cdot, y)$ is $\preceq_L^S$-monotonic, (2) $\mathcal{O}_l(x, \cdot)$ is $\preceq_L^S$-anti-monotonic, (3) $\mathcal{O}_u(x, \cdot)$ is $\preceq_L^H$-monotonic, and (4) $\mathcal{O}_u(\cdot, y)$ is $\preceq_L^H$-anti-monotonic. Consider now some $x_1, x_2, y_1, y_2 \in \mathcal{L}$ s.t. $x_1 \leq x_2$ and $y_2 \leq y_1$, i.e. $(x_1, y_1) \leq_i (x_2, y_2)$. With (1),

$\mathcal{O}_l(x_1,y_2) \preceq^S_L \mathcal{O}_l(x_2,y_2)$. With (2), $\mathcal{O}_l(x_1,y_1) \preceq^S_L \mathcal{O}_l(x_1,y_2)$. With transitivity, $\mathcal{O}_l(x_1,y_1) \preceq^S_L \mathcal{O}_l(x_2,y_2)$. The case for the upper bound (namely, $\mathcal{O}_u(x_2,y_2) \preceq^H_L \mathcal{O}_u(x_1,y_1)$) is similar. □

Remark 11. By Lemma 2, we see that for a symmetric ndao $\mathcal{O}$, $\mathcal{O}_l(\cdot,z)$ is both $\preceq^S_L$-monotonic and $\preceq^H_L$-monotonic, and $\mathcal{O}_l(z,\cdot)$ is both $\preceq^S_L$-anti-monotonic and $\preceq^H_L$-anti-monotonic. This follows immediately from the fact that since $\mathcal{O}$ is symmetric, $\mathcal{O}_l(x,y)=\mathcal{O}_u(y,x)$ for any $x,y\in\mathcal{L}$.

The last remark means that for symmetric operators, the $\preceq^A_i$-monotonicity reduces to a far simpler order, obtained by comparing both the lower and upper bounds as follows: one set X is smaller than another set Y, if for every $x\in X$ there is a $\leq$-larger element in Y, and for every $y\in Y$ there is a $\leq$-smaller element in X.

It seems that the most commonly arising useful non-deterministic approximation operators are symmetric. However, there are also useful non-symmetric ndaos. In the remainder of this section, we provide an example of a non-symmetric ndao that approximates $IC_{\mathcal{P}}$. This operator is inspired by the ultimate semantics for normal (non-disjunctive) logic programs introduced by Denecker, Marek and Truszczyński [22]. First, we recall the ultimate semantics for normal (non-disjunctive) logic programs:

Definition 14. Given a normal logic program $\mathcal{P}$, we define[14]:

$$\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)=\bigcap_{x\subseteq z\subseteq y}\{\alpha\mid\alpha\leftarrow\phi\in\mathcal{P}\text{ and }z(\phi)=\mathsf{T}\},$$
$$\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)=\bigcup_{x\subseteq z\subseteq y}\{\alpha\mid\alpha\leftarrow\phi\in\mathcal{P}\text{ and }z(\phi)=\mathsf{T}\}.$$

The *ultimate approximation operator* is then defined in [22] by:

$$\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}(x,y)=(\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y),\ \mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)).$$

To generalize this operator to an ndao, we proceed as follows: we start by generalizing the idea behind $\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}$ to an operator gathering the heads of rules that are true in every interpretation z in the interval $[x,y]$:

$$\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)=\bigcap_{x\subseteq z\subseteq y}HD_{\mathcal{P}}(z).$$

The immediate consequence operator is then defined as usual, that is: by taking all interpretations that only contain atoms in $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)$ and contain at least one member of every head $\Delta\in\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)$:

$$\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)=\{z\subseteq\bigcup\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)\mid\forall\Delta\in\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)\neq\emptyset:\ z\cap\Delta\neq\emptyset\}.$$

The upper bound operator is constructed entirely analogously, but now gathering the heads of rules that are true in at least one interpretation in $[x,y]$:

$$\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)=\bigcup_{x\subseteq z\subseteq y}HD_{\mathcal{P}}(z)$$

$\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}$ is defined in an identical way to $\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}$, by just replacing $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)$ by $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)$. Finally, the DMT-ndao is defined as:

$$\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}(x,y)=(\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y),\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)).$$

Notice that for a non-disjunctive program $\mathcal{P}$, $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)=\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)$, which also coincides with the deterministic operator defined in Definition 14.

We observe that $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$ is an ndao that approximates $IC_{\mathcal{P}}$:

Proposition 4. *For any disjunctive logic program $\mathcal{P}$, $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$ is an ndao that approximates $IC_{\mathcal{P}}$.*

Notice that the operators $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y)$ and $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y)$ are only defined for consistent interpretations (x,y), and thus $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$ is not a symmetric operator.

Example 9. Consider again the program $\mathcal{P}=\{p\vee q\leftarrow\neg q\}$ from Example 7. Then $\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}$ behaves as follows:

[14] We use the abbreviation DMT for Denecker, Marek and Truszczyński to denote this operator, as to not overburden the use of $\mathcal{IC}^{\mathcal{U}}_{\mathcal{P}}$. Indeed, we will later see that the ultimate operator for non-disjunctive logic programs generalizes to an ndao that is different from the ultimate operator $\mathcal{IC}^{\mathcal{U}}_{\mathcal{P}}$.

- If $q \in y$ then $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y) = \emptyset$ and thus $\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y) = \{\emptyset\}$.
- If $q \notin y$ then $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y) = \{\{p,q\}\}$ and thus $\mathcal{IC}^{\mathsf{DMT},l}_{\mathcal{P}}(x,y) = \{\{p\},\{q\},\{p,q\}\}$.

$\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}$ behaves as follows:

- If $q \in x$ then $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y) = \emptyset$ and thus $\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y) = \emptyset$.
- If $q \notin x$ then $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y) = \{\{p,q\}\}$ and thus $\mathcal{IC}^{\mathsf{DMT},u}_{\mathcal{P}}(x,y) = \{\{p\},\{q\},\{p,q\}\}$.

As an example of the difference between $\mathcal{IC}_{\mathcal{P}}$ and $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$, a normal logic program suffices [22]: consider $\mathcal{P} = \{p \leftarrow p;\ \ p \leftarrow \neg p\}$. Then $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}(\emptyset,\{p\}) = (\{p\},\{p\})$ whereas $\mathcal{IC}_{\mathcal{P}}(\emptyset,\{p\}) = (\emptyset,\{p\})$. This means, among others, that $(\{p\},\{p\})$ is a stable fixpoint of $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$ but not of $\mathcal{IC}_{\mathcal{P}}$.

4. Theory of non-deterministic AFT

We now develop a general theory of approximation of non-deterministic operators. First, in Section 4.1 we show that the notion of consistency from deterministic AFT [23] can be generalized to the non-deterministic setting and holds for any ndao. Then, in Section 4.2 we introduce fixpoint semantics for ndaos. It is shown that in general such a semantics does not preserve the uniqueness or existence properties of Kripke-Kleene semantics from the deterministic setting. In Section 4.3 we consider Kripke-Kleene states that do not have these shortcomings.

4.1. Consistency of approximations

Recall that a *pair* (x,y) is consistent if $x \leq y$, i.e., if it approximates at least one element z ($x \leq z \leq y$). Consistency of a deterministic approximating *operator* means that a consistent input, i.e. an input that approximates at least one element, gives rise to a consistent output, i.e. an output that approximates at least one element. For deterministic operators, consistency of any approximation operator is guaranteed [23, Theorem 9].

This intuition is generalized straightforwardly to non-deterministic approximation operators by requiring that whenever the operator is applied to a consistent pair, there is at least one lower bound which is smaller than at least one upper bound, i.e., there is at least one element approximated by the operator. Formally, this comes down to the following definition:

Definition 15. Given a lattice $L = \langle \mathcal{L}, \leq \rangle$ and an ndao $\mathcal{O}$ on L, we say that $\mathcal{O}$ is *consistent* if for every $x, y \in \mathcal{L}$ with $x \leq y$, there is some $(w,z) \in \mathcal{O}(x,y)$ with $w \leq z$.[15]

We now show the consistency of every ndao.

Proposition 5. *Any ndao $\mathcal{O}$ is consistent.*

Proof. Consider some $x, y \in \mathcal{L}$ s.t. $x \leq y$. Then clearly, $(x,y) \leq_i (x,x)$ and thus, with $\preceq^A_i$-monotonicity of $\mathcal{O}$ and Lemma 1, $\mathcal{O}(x,y) \preceq^S_i \mathcal{O}(x,x)$. Since $\mathcal{O}(x,x) = (\mathcal{O}_l(x,x), \mathcal{O}_u(x,x))$ (in view of the exactness of $\mathcal{O}$), for any $w \in \mathcal{O}_l(x,x)$, we have that $(w,w) \in \mathcal{O}(x,x)$. Thus, since $\mathcal{O}(x,y) \preceq^S_i \mathcal{O}(x,x)$, there is some $(z_1,z_2) \in \mathcal{O}(x,y)$ s.t. $(z_1,z_2) \leq_i (w,w)$, i.e. $z_1 \leq w \leq z_2$. □

For symmetric operators, a stronger notion of consistency holds, namely for every $x \leq y$, it holds that: $\mathcal{O}_l(x,y) \preceq^S_L \mathcal{O}_u(x,y)$ and $\mathcal{O}_l(x,y) \preceq^H_L \mathcal{O}_u(x,y)$. Intuitively, this means that for *every* upper bound we can find a lower bound below the upper bound in question, and likewise, for every lower bound we can find an upper bound above the lower bound in question. Thus, in symmetric operators, *every* lower bound respectively upper bound approximates an element.

Proposition 6. *Let a symmetric ndao $\mathcal{O}$ be given. Then for every $x \leq y$, $\mathcal{O}_l(x,y) \preceq^S_L \mathcal{O}_u(x,y)$ and $\mathcal{O}_l(x,y) \preceq^H_L \mathcal{O}_u(x,y)$.*

Proof. Since $x \leq y$, it holds that $(x,y) \leq_i (x,x)$. Thus, $\mathcal{O}_l(x,y) \preceq^S_L \mathcal{O}_l(x,x)$ and (since $\mathcal{O}_u(x,y) = \mathcal{O}_l(y,x)$), $\mathcal{O}_u(x,x) \preceq^S_L \mathcal{O}_u(x,y)$. Since $\mathcal{O}_l(x,x) = \mathcal{O}_u(x,x)$, we obtain $\mathcal{O}_l(x,y) \preceq^S_L \mathcal{O}_u(x,y)$. The other case is similar. □

Notice that for non-symmetric operators, the above result might not hold, i.e., there might be lower bounds $x_1 \in \mathcal{O}_l(x,y)$ with no correspondent upper bound $y_1 \geq x_1$ in $\mathcal{O}_u(x,y)$, and vice versa. The reason for this is that $\preceq^A_i$-monotonicity only tells us that $\mathcal{O}_l(x,y) \preceq^S_L \mathcal{O}_l(x,x)$ and $\mathcal{O}_u(x,x) \preceq^H_L \mathcal{O}_u(x,y)$. This is demonstrated by the following example:

[15] To avoid clutter, we abuse the notation and write $(w,z) \in \mathcal{O}(x,y)$ to denote that $w \in \mathcal{O}_l(x,y)$ and $z \in \mathcal{O}_u(x,y)$.

Example 10. Consider an ndao $\mathcal{O}$ over $\mathcal{FOUR}$, defined as follows:

	$\mathcal{O}_l(x,y)$	$\mathcal{O}_u(x,y)$
when $x = y$	$\{\mathsf{T}\}$	$\{\mathsf{T}\}$
when $x \neq y$	$\{\mathsf{F},\mathsf{T}\}$	$\{\mathsf{T}\}$

It can be easily observed that $\mathcal{O}$ is a non-deterministic approximation operator: Exactness is clear, and, as $\{\mathsf{F},\mathsf{T}\} \preceq_L^S \{\mathsf{T}\}$, $\preceq_i^A$-monotonicity is also immediate. However, for the consistent pair (U,T), $\mathcal{O}_l(\mathsf{U},\mathsf{T}) = \{\mathsf{F},\mathsf{T}\} \not\preceq_L^H \mathcal{O}_u(\mathsf{U},\mathsf{T}) = \{\mathsf{T}\}$. As $\mathsf{F} \not\leq_i \mathsf{T}$, there is no upper bound in $\mathcal{O}_u(\mathsf{U},\mathsf{T}) = \{\mathsf{T}\}$ corresponding to the lower bound F.

4.2. Fixpoint semantics and Kripke-Kleene interpretations

As its name suggests, a primary goal of AFT is to provide fixpoints of operators and their approximations. In the context of deterministic AFT, fixpoints of an approximation operator are approximations for which applying the approximation operator to the lower and upper bound (x,y) give rise to exactly the same upper and lower bound. Furthermore, Denecker, Marek and Truszczyński [25] show that a unique $\leq_i$-least consistent fixpoint of an approximation operator $\mathcal{O}$ exists and can be constructed by iterating $\mathcal{O}$, starting from $(\bot,\top)$. This fixpoint is termed the *Kripke-Kleene* fixpoint. In this section, we look at fixpoint semantics for non-deterministic approximation operators, and show that existence and uniqueness of a $\leq_i$-least consistent fixpoint of an ndao are not preserved in the non-deterministic setting. Nevertheless, we will show the usefulness of such fixpoints for, e.g. representing the weakly supported semantics.

Recall that an ndao generates, on the basis of a lower bound x and an upper bound y, a set of lower bounds $\{x_1, x_2, \dots\}$ and a set of upper bounds $\{y_1, y_2, \dots\}$. We can then generalize the notion of fixpoints of deterministic approximation operators by stating that (x,y) is a fixpoint of the ndao $\mathcal{O}$ if the "input" lower bound x and the "input" upper bound y are among the "output" lower bounds respectively "output" upper bounds generated on the basis of (x,y).

The idea underlying the Kripke-Kleene fixpoint as being minimally informative can then be directly taken over from the deterministic setting and imposed by definition. Accordingly, we can define fixpoints, and Kripke-Kleene interpretations, of an ndao $\mathcal{O}$ as follows:

Definition 16. Given an ndao $\mathcal{O}$ over $L = \langle \mathcal{L}, \leq \rangle$ and some $x, y \in \mathcal{L}$:

- (x,y) is a *fixpoint* of $\mathcal{O}$, if $x \in \mathcal{O}_l(x,y)$ and $y \in \mathcal{O}_u(x,y)$ (or, somewhat abusing the notation, if $(x,y) \in \mathcal{O}(x,y)$),
- (x,y) is a *Kripke-Kleene interpretation* of $\mathcal{O}$, if it is $\leq_i$-minimal among the fixpoints of $\mathcal{O}$.

Thus, Kripke-Kleene interpretations retain the type of Kripke-Kleene semantics in deterministic AFT, namely pairs of single elements (x,y). Intuitively, these interpretations represent approximations (x,y) of elements such that, when making the "right choices" within the set $\mathcal{O}(x,y)$, $\mathcal{O}$ allows to derive exactly the same lower and upper bound.

The next example shows that uniqueness is no longer guaranteed in the non-deterministic case.

Example 11. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow\}$ from Example 1. There are *two* $\leq_i$-minimal consistent fixpoints of $\mathcal{IC}_\mathcal{P}$ (Definition 12): $(\{p\},\{p,q\})$ and $(\{q\},\{p,q\})$. This is easy to verify, as $\mathcal{IC}_\mathcal{P}(x,y) = (\{\{p\},\{q\},\{p,q\}\}, \{\{p\},\{q\},\{p,q\}\})$ for any $x,y \subseteq \{p,q\}$. We thus see there is no unique $\leq_i$-minimal fixpoint.

We also note that this program has three weakly supported models: $\{p\}$, $\{q\}$ and $\{p,q\}$ that correspond to the total consistent fixpoints $(\{p\},\{p\})$, $(\{q\},\{q\})$ and $(\{p,q\},\{p,q\})$. We shall see in Theorem 1 that this is no coincidence.

The following example shows that existence of a consistent fixpoint of an ndao is not guaranteed either:

Example 12. Since we are interested in a consistent fixpoint, it suffices to restrict our attention to the consistent pairs. Consider an operator $\mathcal{O}$ over the bilattice constructed on the powerset of $\{p,q\}$ and defined as follows:

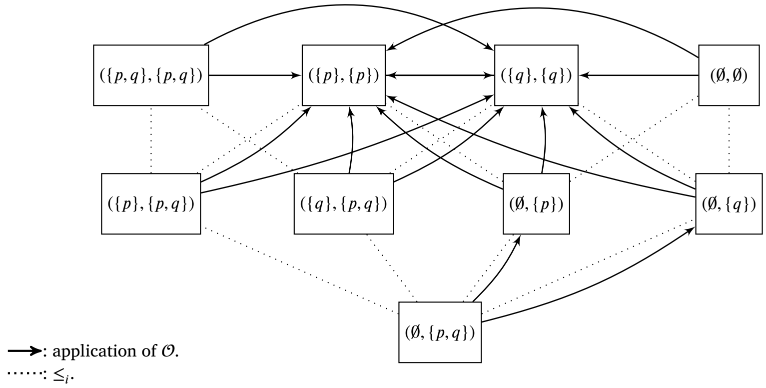
$$\mathcal{O}(\emptyset,\{p,q\}) = (\{\emptyset\},\{\{p\},\{q\}\})$$

$$\mathcal{O}(\{p\},\{p\}) = (\{q\},\{q\})$$

$$\mathcal{O}(\{q\},\{q\}) = (\{p\},\{p\})$$

$$\mathcal{O}(x,y) = (\{\{p\},\{q\}\},\{\{p\},\{q\}\}) \quad \forall (x,y) \notin \{(\{p\},\{p\}),(\{q\},\{q\})\}$$

It is easily observed that $\mathcal{O}$ is $\leq_i$-monotonic. Since $\mathcal{O}_l(x,x) = \mathcal{O}_u(x,x)$ for any $x \subseteq \{p,q\}$, it is also exact. This operator can be visualized as follows:



It can be verified that this operator is $\preceq_i^A$-monotonic, yet it admits no consistent fixpoint.

Although some properties of fixpoints of approximation operators do not carry over from the deterministic to the non-deterministic setting, fixpoints of a non-deterministic operator $\mathcal{O}$ have been studied in the literature. The following theorem (the proof of which appears in Appendix A) shows that in the context of disjunctive logic programming, the fixpoints of the operator $\mathcal{IC}_{\mathcal{P}}$ characterize the weakly supported models of $\mathcal{P}$. It thus provides a first representation of semantics of logic programs that are not covered by (fixpoints of) deterministic AFT.

Theorem 1. *Given a dlp $\mathcal{P}$ and a consistent interpretation $(x,y) \in (\wp(\mathcal{A}_{\mathcal{P}}))^2$, it holds that (x,y) is a weakly supported model of $\mathcal{P}$ iff $(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$.*

To summarize the results in this section, we conclude that in contradistinction to deterministic AFT, an $\leq_i$-minimal fixpoint of an ndao is neither guaranteed to be unique nor guaranteed to exist. Nevertheless, fixpoint semantics allow for an operator-based characterization of weakly supported models of disjunctive logic programs. In the next section we introduce the state semantics which is guaranteed to exist and be unique.

4.3. Kripke-Kleene states

In the previous section, we saw that an (unique) $\leq_i$-minimal fixpoint of an ndao is not guaranteed to exist. This is perhaps not surprising, as the application of an ndao to a pair (x,y) gives rise to a set of lower and upper bounds instead of a single lower and a single upper bound. This means that the method for constructing a $\leq_i$-minimal fixpoint of a deterministic approximation fixpoint operator $\mathcal{O}_{\mathsf{det}}$ over a complete lattice by iteratively constructing more and more precise approximations by the converging sequence

$$(\bot,\top) \leq_i \mathcal{O}_{\mathsf{det}}(\bot,\top) \leq_i \mathcal{O}^2_{\mathsf{det}}(\bot,\top) \leq_i \dots$$

(where $\bot$ and $\top$ respectively represent the minimal and the maximal lattice element) is not well-defined, as an ndao $\mathcal{O}$ cannot be applied to the pair of sets $\mathcal{O}^i(\bot,\top)$. However, we can circumvent this deficit by looking at operators which allow for pairs of sets in their input. Intuitively, we start with a set of lower bounds $\{x_1,x_2,\dots\}$ and a set of upper bounds $\{y_1,y_2,\dots\}$, and construct a new, more precise set of lower respectively upper bounds on the basis of them. Thus, instead of approximating a single element z by a pair of elements (x,y), we are now approximating a set of elements $\{z_1,z_2,\dots\}$ by a pair of sets of elements $\{x_1,x_2,\dots\}$ and $\{y_1,y_2,\dots\}$. An approximation of such a set of elements can then be seen as a *convex set* (Remark 8), bounded below by the lower bounds $\{x_1,x_2,\dots\}$ and bounded above by the upper bounds $\{y_1,y_2,\dots\}$. We first recall some basic notions and notations concerning convex sets.

4.3.1. Preliminaries on convex sets

Recall from Remark 8 that a convex set is a set without "holes", that is, a set X such that if two elements x and y are in X, then also any element between x and y is in X. Such a set can be viewed as consisting of all elements in between a lower bounds and an upper bound. A convex set can then be obtained by the upwards closure of their lower bound and the downwards closure of their upper bound, as defined next:

Definition 17. Given a lattice $L = \langle \mathcal{L}, \leq \rangle$ and an element $x \in \mathcal{L}$, then:

- the *upwards closure* of x is $x\uparrow := \{y \in \mathcal{L} \mid x \leq y\}$.
- the *downwards closure* of x is $x\downarrow := \{y \in \mathcal{L} \mid x \geq y\}$.

We lift this to sets of elements $X \subseteq \mathcal{L}$ as follows:

- the *upwards closure* of X is $X\uparrow := \bigcup_{x \in X} x\uparrow$,
- the *downwards closure* of X is $X\downarrow := \bigcup_{x \in X} x\downarrow$.

A set is *upwards (respectively downwards) closed*, if $X = X\uparrow$ (respectively $X = X\downarrow$). We denote the set of upwards closed (respectively downwards closed) subsets of $\mathcal{L}$ by $\wp_\uparrow(\mathcal{L})$ (respectively $\wp_\downarrow(\mathcal{L})$).

It can be shown that every set of elements is $\preceq^S_L$-equivalent to its upwards closure and $\preceq^H_L$-equivalent to its downwards closure:

Lemma 3. *Given a lattice $L = \langle \mathcal{L}, \leq \rangle$ and a set $X \subseteq \mathcal{L}$, it holds that:*

1. *$X\uparrow \preceq^S_L X$ and $X \preceq^S_L X\uparrow$, and*
2. *$X\downarrow \preceq^H_L X$ and $X \preceq^H_L X\downarrow$.*

Proof. Item 1. Since $X\uparrow \supseteq X$, it immediately holds that $X\uparrow \preceq^S_L X$. We now show $X \preceq^S_L X\uparrow$. Consider some $x \in X\uparrow$. This means there is some $y \in X$ s.t. $y \leq x$.

Item 2. Since $X\downarrow \supseteq X$, it immediately holds that $X \preceq^H_L X\downarrow$. We now show $X\downarrow \preceq^H_L X$. Consider some $x \in X\downarrow$. By the definition of $X\downarrow$, there is some $y \in X$ s.t. $x \leq y$. □

The last lemma gives further motivation to using $\preceq^S_L$ for comparing lower bounds and $\preceq^H_L$ for comparing upper bounds. Indeed, if $x \in X$ is a lower bound for z, then any y such that $x \leq y$ is also a good candidate for being z (or, alternatively, approximating z "from below"), and likewise, if $x \in X$ is an upper bound for z, then any y such that $y \leq x$ is also a good candidate for being z (or, alternatively, approximating z "from above").

Lemma 3 means that we can restrict attention to upwards closed sets when using $\preceq^S_L$ and to downwards closed sets when using $\preceq^H_L$, without losing any information.

The sets $\wp_\uparrow(\mathcal{L})$ and $\wp_\downarrow(\mathcal{L})$ admit greatest lower bounds under $\preceq^S_L$ and least upper bounds under $\preceq^H_L$ (respectively):

Lemma 4. *Let $\mathcal{X} \subseteq \wp_\uparrow(\mathcal{L})$ be given. Then $\bigcup \mathcal{X}$ is the greatest lower bound (under $\preceq^S_L$) of $\mathcal{X}$ and $\bigcap \mathcal{X}$ is the least upper bound (under $\preceq^S_L$) of $\mathcal{X}$.*[16]

Proof. We show that $\bigcup \mathcal{X}$ is the greatest lower bound by showing the following claims:

- $\bigcup \mathcal{X} \in \wp_\uparrow(\mathcal{L})$. To see this, consider some $x \in \bigcup \mathcal{X}$. Then there is some $X \in \mathcal{X}$ s.t. $x \in X$, and thus since $X \in \wp_\uparrow(\mathcal{L})$, every $y \in X$ s.t. $x \leq y$ is also in X, and therefore in $\bigcup \mathcal{X}$.
- $\bigcup \mathcal{X} \preceq^S_L X$ for any $X \in \mathcal{X}$. This immediately follows from the fact that for any $x \in X$, $x \in \bigcup \mathcal{X}$.
- For any $Z \in \wp_\uparrow(\mathcal{L})$ s.t. $Z \preceq^S_L X$ for every $X \in \mathcal{X}$, $Z \preceq^S_L \bigcup \mathcal{X}$. For this, consider some $Z \in \wp_\uparrow(\mathcal{L})$ s.t. $Z \preceq^S_L X$ for every $X \in \mathcal{X}$ and consider some $x \in \bigcup \mathcal{X}$. Since $x \in X$ for some $X \in \mathcal{X}$, there is some $z \in Z$ s.t. $z \leq x$. Thus, for every $x \in \bigcup \mathcal{X}$, there is some $z \in Z$ s.t. $z \leq x$, which implies that $Z \preceq^S_L \bigcup \mathcal{X}$.

We show that $\bigcap \mathcal{X}$ is the least upper bound by showing the following claims:

- $\bigcap \mathcal{X} \in \wp_\uparrow(\mathcal{L})$. To see this, consider some $x \in \bigcap \mathcal{X}$. Then for every $X \in \mathcal{X}$, it holds that $x \in X$, and thus since $X \in \wp_\uparrow(\mathcal{L})$, every $y \in X$ s.t. $x \leq y$ is also in X, and therefore (since this holds for every $X \in \mathcal{X}$), y is also in $\bigcap \mathcal{X}$.
- $X \preceq^S_L \bigcap \mathcal{X}$ for any $X \in \mathcal{X}$. This follows from the fact that for any $x \in \bigcap \mathcal{X}$, $x \in X$ for every $X \in \mathcal{X}$.
- For any $Z \in \wp_\uparrow(\mathcal{L})$ s.t. $X \preceq^S_L Z$ for every $X \in \mathcal{X}$, $\bigcap \mathcal{X} \preceq^S_L Z$. For this, consider some $Z \in \wp_\uparrow(\mathcal{L})$ s.t. $X \preceq^S_L Z$ for every $X \in \mathcal{X}$ and consider some $z \in Z$. Since $X \preceq^S_L Z$, there is some $x \in X$ s.t. $x \leq z$. Since X is upwards closed, $z \in X$. Notice that this holds for any $X \in \mathcal{X}$, and thus $z \in \bigcap \mathcal{X}$, which means that $Z \subseteq \bigcap \mathcal{X}$ and thus $\bigcap \mathcal{X} \preceq^S_L Z$. □

The next lemma is proven like the previous one.

Lemma 5. *Let some $\mathcal{X} \subseteq \wp_\downarrow(\mathcal{L})$ be given. Then $\bigcap \mathcal{X}$ is the greatest lower bound (under $\preceq^H_L$) of $\mathcal{X}$ and $\bigcup \mathcal{X}$ is the least upper bound (under $\preceq^H_L$) of $\mathcal{X}$.*

By the last two lemmas, one can construct lower respectively upper bounds for sets of pairs of sets under $\preceq^A_i$:

[16] Recall that we use small letters to denote elements of lattice, capital letters to denote sets of elements, and capital calligraphic letters to denote sets of sets of elements (Table 1).

Corollary 1. *Let some $\mathcal{Z} \subseteq \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$ be given. Then $(\bigcup\{X \mid (X,Y) \in \mathcal{Z}\}, \bigcup\{Y \mid (X,Y) \in \mathcal{Z}\})$ is the greatest lower bound of (under $\preceq_i^A$) $\mathcal{Z}$ and $(\bigcap\{X \mid (X,Y) \in \mathcal{Z}\}, \bigcap\{Y \mid (X,Y) \in \mathcal{Z}\})$ is the least upper bound of (under $\preceq_i^A$) of $\mathcal{Z}$.*

To make things a bit more graspable, we illustrate the above results with an example.

Example 13. Consider the lattice $\langle \wp(\{p,q\}), \subseteq\rangle$.

- By Lemma 4, the glb under $\preceq_L^S$ of the upwards closed sets $\{\{p\},\{q\},\{p,q\}\}$, $\{\{q\},\{p,q\}\}$ and $\{\{p,q\}\}$ is

$$\{\{p\},\{q\},\{p,q\}\} \cup \{\{q\},\{p,q\}\} \cup \{\{p,q\}\} = \{\{p\},\{q\},\{p,q\}\}.$$

- By Lemma 5, the $\preceq_L^H$-glb of the downwards closed sets $\{\{p\},\{q\},\emptyset\}$, $\{\{p\},\emptyset\}$ is

$$\{\{p\},\{q\},\emptyset\} \cap \{\{p\},\emptyset\} = \{\{p\},\emptyset\}.$$

- Consider the following pairs of upwards and downwards closed sets:

$$(X_1, Y_1) = (\{\{p\},\{p,q\}\}, \{\emptyset,\{p\},\{q\},\{p,q\}\})$$
$$(X_2, Y_2) = (\{\{p,q\}\}, \{\emptyset,\{p\},\{q\},\{p,q\}\})$$
$$(X_3, Y_3) = (\{\{q\},\{p,q\}\}, \{\emptyset,\{q\}\})$$

According to Corollary 1, the $\preceq_i^A$-glb and $\preceq_i^A$-lub of $\{(X_1, Y_1),(X_2, Y_2),(X_3, Y_3)\}$ are obtained as follows:

$$glb_{\preceq_i^A}(\{(X_1, Y_1),(X_2, Y_2),(X_3, Y_3)\}) = (\bigcup_{i=1}^{3} X_i, \bigcup_{i=1}^{3} Y_i) = (\{\{p\},\{q\},\{p,q\}\}, \{\emptyset,\{p\},\{q\},\{p,q\}\})$$
$$lub_{\preceq_i^A}(\{(X_1, Y_1),(X_2, Y_2),(X_3, Y_3)\}) = (\bigcap_{i=1}^{3} X_i, \bigcap_{i=1}^{3} Y_i) = (\{\{p,q\}\}, \{\emptyset\})$$

In Fig. 2, the above pairs of sets are visualized as convex sets. In more detail, the elements of convex sets are highlighted gray. For example, (X_1,Y_1) corresponds to the convex set $\{\{p\},\{p,q\}\}$ and therefore the elements $\{p\}$ and $\{p,q\}$ are highlighted in Fig. 2.

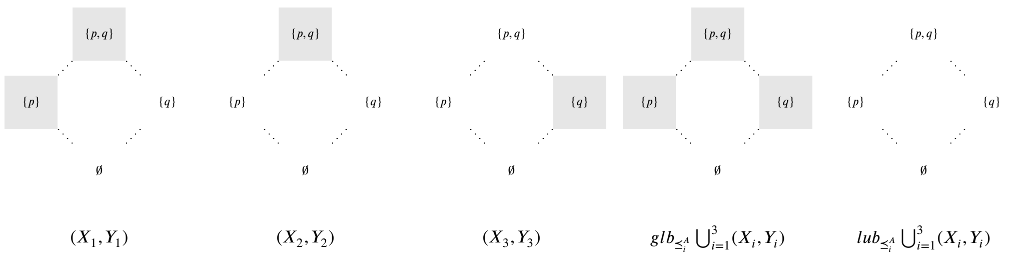


Fig. 2. Visualization of convex sets in Example 13, with elements of convex sets highlighted in gray.

It is interesting to note that the construction described above does *not* depend on the completeness of the original lattice $\mathcal{L}$. In more detail, Lemmas 4 and 5 and Corollary 1 mean that $\langle \wp_\uparrow(\mathcal{L}), \preceq_L^S\rangle$, $\langle \wp_\downarrow(\mathcal{L}), \preceq_L^H\rangle$ and $\langle \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L}), \preceq_i^A\rangle$ are complete lattices, even if $\langle \mathcal{L}, \leq\rangle$ is not a complete lattice. To illuminate this somewhat surprising fact, consider the following example:

Example 14. Consider the lattice $\langle \mathbb{N}, \leq\rangle$. As $\mathbb{N}$ has no lub, this lattice is not complete. Now, let $\{\{i \mid i \leq j\} \mid j \in \mathbb{N}\}$. Then the $\preceq_L^H$-lub of $\{\{i \mid i \leq j\} \mid j \in \mathbb{N}\}$ is $\mathbb{N}$. Thus, every set in $\langle \wp_\downarrow(\mathbb{N}), \preceq_L^H\rangle$ has a least upper bound, even though $\langle \mathbb{N}, \leq\rangle$ does not. Likewise, every set in $\langle \wp_\uparrow(\mathbb{N}), \preceq_L^S\rangle$ and in $\langle \wp_\uparrow(\mathbb{N}) \times \wp_\downarrow(\mathbb{N}), \preceq_i^A\rangle$ has a least upper bound.

In general, $(\mathcal{L},\mathcal{L})$ is the $\preceq_i^A$-glb of $\wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$ whereas $(\emptyset,\emptyset)$ is the $\preceq_i^A$-lub of $\wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$. It is interesting to note that, when $\langle \mathcal{L}, \leq\rangle$ is a complete lattice, $(\bot\uparrow, \top\downarrow) = (\mathcal{L},\mathcal{L})$, i.e. the $\preceq_i^A$-glb of $\wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$ corresponds to the convex set obtained on the basis of the $\leq_i$-glb $(\bot,\top)$ of $\mathcal{L}^2$ (and likewise for $(\emptyset,\emptyset)$).

4.3.2. Non-deterministic state operators and their fixpoints

We can now introduce *non-deterministic state approximation operators*[17] that generate a convex set on the basis of a convex set. Thus, the type of a non-deterministic state approximation operator $\mathcal{O}'$ is $\wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L}) \to \wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L})$. As outlined above, the idea is that a non-deterministic state approximation operator generates an approximation of a set of elements $\{z_1, z_2, \ldots\}$ on the basis of an approximation of a set of elements. Just like an ndao, we require information-monotonicity, i.e. more precise inputs give rise to more precise outputs, and exactness, i.e., a convex set consisting of a single element as input gives rise to at least one element being both in the generated lower and upper bound.

Definition 18. Let a lattice $L = \langle \mathcal{L}, \leq \rangle$ be given. Then an operator $\mathcal{O}' : \wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L}) \to \wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L})$ is a *non-deterministic state operator* (in short, ndso) if it satisfies the following properties:

- $\mathcal{O}'$ is $\preceq_i^A$-monotonic, and
- $\mathcal{O}'$ is *exact*, i.e. for every $x \in \mathcal{L}$, there is some $z \in \mathcal{L}$ such that $z \in \mathcal{O}'_l(\{x\}, \{x\}) \cap \mathcal{O}'_u(\{x\}, \{x\})$.[18]

An ndso *approximates* a non-deterministic operator O iff for any $x \in \mathcal{L}$, $\mathcal{O}'(\{x\}, \{x\}) = (O(x)\uparrow, O(x)\downarrow)$.

Remark 12. A non-deterministic state operator can be straightforwardly derived on the basis of an ndao as follows (but there are also other ways to do this, see e.g. Definition 23):

$$\mathcal{O}'(\mathbf{X}) = \left(\bigcup_{(x,y)\in\mathbf{X}} \mathcal{O}_l(x,y)\uparrow, \bigcup_{(x,y)\in\mathbf{X}} \mathcal{O}_u(x,y)\downarrow \right)$$

In that case we say $\mathcal{O}'$ is *derived from* $\mathcal{O}$. If $\mathcal{O}$ approximates O then so does $\mathcal{O}'$. Furthermore, $\mathcal{O}'$ is $\preceq_i^A$-monotonic. In fact, the $\preceq_i^A$-monotonicity of $\mathcal{O}'$ is independent of the $\preceq_i^A$-monotonicity of $\mathcal{O}$, as we see in the following proposition:

Proposition 7. *For any operator $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$, it holds that $\mathcal{O}' : \wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L}) \to \wp_{\uparrow}(\mathcal{L}) \times \wp_{\downarrow}(\mathcal{L})$ is $\preceq_i^A$-monotonic.*

Proof. Consider some $X, X', Y, Y' \subseteq \mathcal{L}$ s.t. $(X, Y) \preceq_i^A (X', Y')$, i.e. $X \preceq_L^S X'$ and $Y' \preceq_L^H Y$. We show that $\mathcal{O}'(X,Y) \preceq_i^A \mathcal{O}'(X',Y')$. Consider first $z \in (\mathcal{O}'(X',Y'))_1$.[19] This means that there are some $x' \in X'\uparrow$ and $y' \in Y'\downarrow$ s.t. $z \in (\mathcal{O}(x',y'))_1\uparrow$. Since $X \preceq_L^S X'$ and $Y' \preceq_L^H Y$, there is some $x \in X$ s.t. $x \leq x'$ and some $y \in Y$ s.t. $y' \leq y$. Thus, $x' \in X\uparrow$ and $y' \in Y\downarrow$, which means that $z \in (\mathcal{O}'(X,Y))_1$. The case for $(\mathcal{O}'(X',Y'))_2$ is similar. We thus have shown that $(\mathcal{O}'(X',Y'))_1 \subseteq (\mathcal{O}'(X,Y))_1$ and $(\mathcal{O}'(X',Y'))_2 \subseteq (\mathcal{O}'(X,Y))_2$, which implies $(\mathcal{O}'(X,Y))_1 \preceq_L^S (\mathcal{O}'(X,Y))_1$ and $(\mathcal{O}'(X',Y'))_2 \preceq_L^H (\mathcal{O}'(X',Y'))_2$, and thus we obtain $\mathcal{O}'(X,Y) \preceq_i^A \mathcal{O}'(X',Y')$. □

Likewise, if one is given an ndso $\mathcal{O}'$, one can obtain an $\preceq_i^A$-monotonic operator $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ by simply letting $\mathcal{O}(x,y) = \mathcal{O}'(\{x\},\{y\})$. Such an operator is not guaranteed to be exact, though.

Example 15. An ndso that approximates $IC_{\mathcal{P}}$ can be defined as follows:

$$\mathcal{IC}'_{\mathcal{P}}(\mathbf{X}) = \left(\bigcup_{(x,y)\in\mathbf{X}} \mathcal{IC}^l_{\mathcal{P}}(x,y)\uparrow, \bigcup_{(x,y)\in\mathbf{X}} \mathcal{IC}^u_{\mathcal{P}}(x,y)\downarrow \right).$$

For instance, in case that $\mathcal{P} = \{p \vee q \leftarrow; \quad r \vee s \leftarrow \neg q\}$, we have:

$$\mathcal{IC}'_{\mathcal{P}}((\{\{p\},\{q\},\{p,q\}\}, \{\emptyset,\{p\},\{q\},\{p,q\}\})) =$$
$$\left(\left(\bigcup_{\emptyset\neq x\subseteq\{p,q\}, y\subseteq\{p,q\}} \mathcal{IC}^l_{\mathcal{P}}(x,y) \right)\uparrow, \left(\bigcup_{\emptyset\neq x\subseteq\{p,q\}, y\subseteq\{p,q\}} \mathcal{IC}^u_{\mathcal{P}}(x,y) \right)\downarrow \right) = (Z_1, Z_1),$$

where $Z_1 = \Big\{ \{p\},\{q\},\{p,q\},\{p,r\},\{q,r\},\{p,s\},\{q,s\},\{p,q,r\},\{p,q,s\} \Big\}$.

We show the following property of non-deterministic state operators:

Lemma 6. *An operator $\mathcal{O}' : \wp(\mathcal{L})^2 \to \wp(\mathcal{L})^2$ is $\preceq_i^A$-monotonic iff for any $X \subseteq \mathcal{L}$, $\mathcal{O}'_l(\cdot, X)$ is $\preceq_L^S$-monotonic, $\mathcal{O}'_l(X,\cdot)$ is $\preceq_L^S$-anti-monotonic, $\mathcal{O}'_u(X,\cdot)$ is $\preceq_L^H$-monotonic, and $\mathcal{O}'_u(\cdot,X)$ is $\preceq_L^H$-anti-monotonic.*

[17] The use of the word *state* comes from disjunctive logic programming, where a set of interpretations is often called a state [44,51].

[18] Alternatively, with a slight abuse of the notations, $(z,z) \in \mathcal{O}'(\{x\},\{x\})$.

[19] Recall that we use $(X,Y)_1$ to denote the first component of the pair (X,Y), i.e. $(X,Y)_1 := X$. Similarly for $(X,Y)_2$.

Proof. [⇒]: Suppose that $\mathcal{O}' : \wp(\mathcal{L})^2 \to \wp(\mathcal{L})^2$ is $\preceq_i^A$-monotonic and consider some $X, Y_1, Y_2 \subseteq \mathcal{L}$ s.t. $Y_1 \preceq_L^S Y_2$. We first show that $\mathcal{O}'_l(Y_1, X) \preceq_L^S \mathcal{O}'(Y_2, X)$. For this, notice that $(Y_1, X) \preceq_i^A (Y_2, X)$ (because $Y_1 \preceq_L^S Y_2$). Since $\mathcal{O}'$ is $\preceq_i^A$-monotonic, $\mathcal{O}'(Y_1, X) \preceq_i^A \mathcal{O}'(Y_2, X)$, which on its turn means that $\mathcal{O}'_l(Y_1, X) \preceq_L^S \mathcal{O}'_l(Y_2, X)$.

We now show that $\mathcal{O}'_u(X, \cdot)$ is $\preceq_L^H$-monotonic. For this, consider some $X, Y_1, Y_2 \subseteq \mathcal{L}$ s.t. $Y_2 \preceq_L^H Y_1$. Then $(X, Y_1) \preceq_i^A (X, Y_2)$ (since $X \preceq_L^S X$). With the $\preceq_i^A$-monotonicity of $\mathcal{O}'$, $\mathcal{O}'(X, Y_1) \preceq_i^A \mathcal{O}'(X, Y_2)$, which implies that $\mathcal{O}'_u(X, Y_2) \preceq_L^H \mathcal{O}'_u(X, Y_1)$. Thus, $Y_2 \preceq_L^H Y_1$ implies $\mathcal{O}'_u(X, Y_2) \preceq_L^H \mathcal{O}'_u(X, Y_1)$ and $\mathcal{O}'_u(X, \cdot)$ is $\preceq_L^H$-monotonic (for any $X \subseteq \mathcal{L}$). The two other cases are similar.

[⇐]: Suppose that for any $X \subseteq \mathcal{L}$, $\mathcal{O}'_l(\cdot, X)$ is $\preceq_L^S$-monotonic, $\mathcal{O}'_l(X, \cdot)$ is $\preceq_L^S$-anti-monotonic, $\mathcal{O}'_u(X, \cdot)$ is $\preceq_L^H$-monotonic, and $\mathcal{O}'_u(\cdot, X)$ is $\preceq_L^H$-anti-monotonic. Consider some $X_1, X_2, Y_1, Y_2 \subseteq \mathcal{L}$ s.t. $(X_1, Y_1) \preceq_i^A (X_2, Y_2)$. We first show that $\mathcal{O}'_l(X_1, Y_1) \preceq_L^S \mathcal{O}'_l(X_2, Y_2)$. For this, notice that $Y_2 \preceq_L^S Y_1$ and thus, with the $\preceq_L^S$-anti-monotonicity of $\mathcal{O}'_l(X_1, \cdot)$, $\mathcal{O}'_l(X_1, Y_1) \preceq_L^S \mathcal{O}'_l(X_1, Y_2)$. Likewise, it can be shown that $\mathcal{O}'_l(X_1, Y_2) \preceq_L^S \mathcal{O}'_l(X_2, Y_2)$ and so $\mathcal{O}'_l(X_1, Y_1) \preceq_L^S \mathcal{O}'_l(X_2, Y_2)$. The proof for $\mathcal{O}'_u$ is analogous. □

We define consistency for an ndso analogously as for an ndao, namely:

Definition 19. An ndso $\mathcal{O}'$ is *consistent* if for every $x, y \in \mathcal{L}$ s.t. $x \leq y$, there is a $w \in \mathcal{O}'_l(\{x\}, \{y\})$ and $z \in \mathcal{O}'_u(\{x\}, \{y\})$, (or, slightly abusing notation, $(w, z) \in \mathcal{O}'(\{x\}, \{y\})$, such that $w \leq z$.

We note that for a consistent ndso $\mathcal{O}'$, for any $X, Y \subseteq \mathcal{L}$ for which $X\uparrow \cap Y\downarrow \neq \emptyset$, it holds that $\mathcal{O}'_l(X, Y))\uparrow \cap \mathcal{O}'_u(X, Y))\downarrow \neq \emptyset$. Therefore, for any consistent ndso, a non-empty convex set in the input (i.e. $X\uparrow \cap Y\downarrow \neq \emptyset$) gives rise to a non-empty convex set in the output $(\mathcal{O}'_l(X, Y))\uparrow \cap \mathcal{O}'_u(X, Y))\downarrow \neq \emptyset)$.

The next proposition shows that state operators are consistent (cf. Proposition 5 for ndao's).

Proposition 8. *Let $\mathcal{O}' : \wp(\mathcal{L})^2 \to \wp(\mathcal{L})^2$ be an ndso. Then $\mathcal{O}'$ is consistent.*

Proof. Suppose that $\mathcal{O}'$ is an ndso, which implies that it is exact, i.e., for every $x \in \mathcal{L}$, there is some $z \in \mathcal{L}$ such that $z \in \mathcal{O}'_l(\{x\}, \{x\})$ and $z \in \mathcal{O}'_u(\{x\}, \{x\})$. This means that there is some $z \in \mathcal{L}$ s.t. $z \in \mathcal{O}'_l(\{x\}, \{x\}) \cap \mathcal{O}'_u(\{x\}, \{x\})$. By Lemma 6, $\mathcal{O}'_l(\{x\}, \{y\}) \preceq_L^S \mathcal{O}'_l(\{x\}, \{x\})$ and $\mathcal{O}'_u(\{x\}, \{x\}) \preceq_L^H \mathcal{O}'_u(\{x\}, \{y\})$. Thus, there is some $w \in \mathcal{O}'_l(\{x\}, \{y\})$ s.t. $w \leq z$ and there is some $w' \in \mathcal{O}'_u(\{x\}, \{y\})$ s.t. $z \leq w'$. With transitivity of $\leq$, $w \leq w'$ and thus (slightly abusing notation) there is a consistent pair $(w, w') \in \mathcal{O}'(x, y)$. □

Another useful property is the fact that the computation of $\mathcal{O}'(X\uparrow, Y\downarrow)$ can be simplified by computing $\mathcal{O}'(X, Y)$:

Lemma 7. *Let $\mathcal{O}' : \wp(\mathcal{L})^2 \to \wp(\mathcal{L})^2$ be an ndso. Then for any $X, Y \subseteq \mathcal{L}$, $\mathcal{O}'(X\uparrow, Y\downarrow) = \mathcal{O}'(X, Y)$.*

Proof. The ⊇-direction is clear. Suppose now that $w \in (\mathcal{O}'(X\uparrow, Y\downarrow))_1$ and $z \in (\mathcal{O}'(X\uparrow, Y\downarrow))_1$. Then there are some $x \in X\uparrow$ and $y \in Y\downarrow$ s.t. $w \in \mathcal{O}_l(x, y)\uparrow$ and $z \in \mathcal{O}_u(x, y)\downarrow$. Thus, there are some $x' \in X$ and some $y' \in Y$ s.t. $x' \leq x$ and $y \leq y'$. This implies that $(x', y') \leq_i (x, y)$ and with the $\preceq_i^A$-monotonicity of $\mathcal{O}$, we have $\mathcal{O}(x', y') \preceq_i^A \mathcal{O}(x, y)$. Thus, $\mathcal{O}_l(x', y') \preceq_L^S \mathcal{O}_l(x, y)$, which implies that $\mathcal{O}'_l(x', y') = \mathcal{O}_l(x', y')\uparrow \supseteq \mathcal{O}_l(x, y)\uparrow$. The case for the upper bound is analogous. □

We now show that an ndso admits a unique $\preceq_i^A$-least fixpoint:

Theorem 2. *Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice. Every $\preceq_i^A$-monotonic operator $\mathcal{O}' : \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L}) \to \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$ admits a unique $\preceq_i^A$-least fixpoint that can be constructed by iterative application of $\mathcal{O}'$ to $(\mathcal{L}, \mathcal{L})$. If $\mathcal{O}'$ is exact, this fixpoint is consistent.*

Proof. Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice and let $\mathcal{O}' : \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L}) \to \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$ be a $\preceq_i^A$-monotonic operator. Then $\langle \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L}), \preceq_i^A \rangle$ forms a complete lattice with $\preceq_i^A$-glb $(\mathcal{L}, \mathcal{L})$ (in view of Corollary 1 and since $\preceq_i^A$ is a reflexive, transitive and anti-symmetric order over $\wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$). We can apply Knaster and Tarski's fixpoint theorem to show that the set of fixpoints of $\mathcal{O}'$ forms a complete lattice, and thus the $\preceq_i^A$-monotonic operator $\mathcal{O}'$ admits a unique $\preceq_i^A$-least fixpoint. Consistency follows from Proposition 8. The fact that the $\preceq_i^A$-least fixpoint can be constructed iteratively follows from Theorem 5.1 shown by Cousot and Cousot [19]. □

We call the $\preceq_i^A$-least fixpoint of $\mathcal{O}'$ that is guaranteed by Theorem 2 the *Kripke-Kleene state of $\mathcal{O}'$*, and denote it by $\mathsf{KK}(\mathcal{O}')$. Some examples of Kripke-Kleene state for concrete logic programs can be found in Examples 16 and 17 below.

For the Kripke-Kleene state of an ndso that is derived from an ndao $\mathcal{O}$, we can show the following additional results:

Theorem 3. *Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice. Given an ndso $\mathcal{O}'$ derived from an ndao $\mathcal{O}$, we have the following:*

- *For any fixpoint (x, y) of $\mathcal{O}$, $\mathsf{KK}(\mathcal{O}') \preceq_i^A (x, y)$,*
- *$\mathsf{KK}(\mathcal{O}')$ contains at least one consistent pair.*
- *If $\mathcal{O}'$ approximates a non-deterministic operator O, then for any $x \in \mathcal{L}$ s.t. $x \in O(x)$, it holds that $\mathsf{KK}(\mathcal{O}') \preceq_i^A (x, x)$.*

Proof. We first show two lemmas:

Lemma 8. *For any* $x, y \in \mathcal{L}$*,* $\mathcal{O}'(x, y) \preceq_i^A \mathcal{O}(x, y)$*.*

Proof. Since $\mathcal{O}'(x, y) = (\mathcal{O}_l(x, y)\uparrow, \mathcal{O}_u(x, y)\downarrow)$, for every $w \in \mathcal{O}_l(x, y)$, there is some $w' \in \mathcal{O}_l(x, y)\uparrow$ s.t. $w' \leq w$ and for every $z \in \mathcal{O}_u(x, y)$ there is some $z' \in \mathcal{O}_u(x, y)\downarrow$ s.t. $z \leq z'$ (namely, w and z themselves). Thus, $\mathcal{O}'_l(x, y) \preceq_L^S \mathcal{O}_l(x, y)$ and $\mathcal{O}_u(x, y) \preceq_L^H \mathcal{O}'_u(x, y)$, so the lemma is obtained. ■

Lemma 9. *For any* $X, Y \subseteq \wp(\mathcal{L})$*,* $x \in X$ *and* $y \in Y$*,* $(X, Y) \preceq_i^A (\{x\}, \{y\})$*.*

Proof. Clearly, there are some $x' \in X$ and $y' \in Y$ (namely $x' = x$ and $y' = y$) s.t. $x' \leq x$ and $y \leq y'$. ■

The proof of Theorem 3 now continues as follows: For the first item, by Proposition 2, we have that $\mathsf{K}(\mathcal{O}') = (\mathcal{O}')^\alpha(\bot, \top)$ for some ordinal α. Furthermore, $(\bot, \top) \leq_i (x, y)$. By the $\preceq_i^A$-monotonicity of $\mathcal{O}$, $\mathcal{O}'(\bot, \top) \preceq_i^A \mathcal{O}'(\{(x, y)\})$. By Lemma 8, $\mathcal{O}'(\{(x, y)\}) \preceq_i^A \mathcal{O}(x, y)$, and by Lemma 9 and since $(x, y) \in \mathcal{O}(x, y)$, $\mathcal{O}'(\bot, \top) \preceq_i^A (x, y)$. We can repeat this process until we reach the ordinal α, and thus $(\mathcal{O}')^\alpha(\bot, \top) = \mathsf{KK}(\mathcal{O}') \preceq_i^A (x, y)$.

The second item is an immediate consequence of Proposition 8.

The proof of the last item is similar to that of the first item. □

Next, we show that for deterministic operators, the Kripke-Kleene state coincides with the Kripke-Kleene fixpoint:

Proposition 9. *Let* $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ *be an ndao on a complete lattice* $\langle \mathcal{L}, \leq \rangle$ *s.t.* $\mathcal{O}(x, y)$ *is a pair of singleton sets for every* $x, y \in \mathcal{L}$*, let* $\mathcal{O}^{\mathsf{AFT}}$ *be defined by* $\mathcal{O}^{\mathsf{AFT}}(x, y) = (w, z)$ *where* $\mathcal{O}(x, y) = (\{w\}, \{z\})$ *and let the Kripke-Kleene fixpoint of* $\mathcal{O}^{\mathsf{AFT}}$ *be given by* $(x^{\mathsf{kk}}, y^{\mathsf{kk}})$*. Then* $\mathsf{KK}(\mathcal{O}') = (x^{\mathsf{kk}}\uparrow, y^{\mathsf{kk}}\downarrow)$*, where* $\mathcal{O}'$ *is obtained from* $\mathcal{O}$ *as described in Remark 12.*

Proof. By Proposition 3, $\mathcal{O}^{\mathsf{AFT}}$ is an approximation operator, and thus admits a Kripke-Kleene fixpoint. For every $x, y \in \mathcal{L}$, it holds that

$$\mathcal{O}'(x, y) = (\{\mathcal{O}_l^{\mathsf{AFT}}(x, y)\}\uparrow, \{\mathcal{O}_u^{\mathsf{AFT}}(x, y)\}\downarrow).$$

Notice furthermore that for any $x, y \in \mathcal{L}$, if $x' \in x\uparrow$ and $y' \in y\downarrow$, $\mathcal{O}(x, y) \preceq_i \mathcal{O}(x', y')$, and so $\mathcal{O}_l(x, y) \preceq_L^S \mathcal{O}_l(x', y')$, i.e., for every $w' \in \mathcal{O}_l(x', y')$ there is a $w \in \mathcal{O}_l(x, y)$ s.t. $w \leq w'$. Thus, $\mathcal{O}_l(x', y') \subseteq \mathcal{O}_l(x, y)\uparrow$ (and similarly for the upper bound: $\mathcal{O}_u(x', y') \supseteq \mathcal{O}_u(x, y)$). This means that for any ordinal α,

$$\mathcal{O}'^{\alpha}(\bot, \top) = (\{(\mathcal{O}_l^{\mathsf{AFT}})^\alpha(\bot, \top)\}\uparrow, \{(\mathcal{O}_u^{\mathsf{AFT}})^\alpha(\bot, \top)\}\downarrow).$$

A simple inductive argument then shows the proposition, as $\mathsf{KK}(\mathcal{O}') = \mathcal{O}'^{\alpha}(\bot, \top)$ for the smallest ordinal α under which a fixpoint is reached, and $(x^{\mathsf{KK}}, y^{\mathsf{KK}}) = (\mathcal{O}^{\mathsf{AFT}})^\alpha(\bot, \top)$ for the smallest ordinal α under which a fixpoint is reached. □

From Proposition 9, we immediately obtain the following corollary in the context of disjunctive logic programs:

Corollary 2. *Given a dlp* $\mathcal{P}$*,* $\mathsf{KK}(\mathcal{IC}_\mathcal{P}) \preceq_i^A (x, y)$ *for every weakly supported model* (x, y) *of* $\mathcal{P}$*.*

Proof. This is immediate from Theorem 3 and since $\mathcal{IC}_\mathcal{P}$ is an ndao (Proposition 2) whose fixpoints coincide with the weakly supported models of $\mathcal{P}$ (Theorem 1). □

Intuitively, the convex set represented by $\mathsf{KK}(\mathcal{IC}_\mathcal{P})$ contains every weakly supported model of $\mathcal{P}$.

We now show some examples for computing Kripke-Kleene states of dlps:

Example 16. Let $\mathcal{P} = \{p \vee q \leftarrow\}$. We calculate $\mathsf{KK}(\mathcal{IC}_\mathcal{P})$ as follows:

- $\mathcal{IC}'_\mathcal{P}(\emptyset, \{p, q\}) = (\{\{p\}, \{q\}, \{p, q\}\}\uparrow, \{\{p\}, \{q\}, \{p, q\}\}\downarrow)$.[20] This can be seen by observing that $\mathcal{IC}^l_\mathcal{P}(\emptyset, \{p, q\}) = \{\{p\}, \{q\}, \{p, q\}\}$ and $\mathcal{IC}^u_\mathcal{P}(\emptyset, \{p, q\}) = \mathcal{IC}^l_\mathcal{P}(\{p, q\}, \emptyset) = \{\{p\}, \{q\}, \{p, q\}\}$.
- $\mathcal{IC}'_\mathcal{P}((\{\{p\}, \{q\}, \{p, q\}\}\uparrow, \{\{p\}, \{q\}, \{p, q\}\}\downarrow)) = (\{\{p\}, \{q\}, \{p, q\}\}\uparrow, \{\{p\}, \{q\}, \{p, q\}\}\downarrow)$ (This can be seen by observing that $\mathcal{IC}^z_\mathcal{P}(x, y) = \{\{p\}, \{q\}, \{p, q\}\}$ for any $z \in \{l, u\}$ and any $x, y \subseteq \{p, q\}$). Thus, a fixpoint is reached.

The Kripke-Kleene state in this case thus corresponds to the convex set $\{\{p\}, \{q\}, \{p, q\}\}$.

[20] Even though this is in principle redundant, we added $\{p, q\}$ to the first component for clarity.

Example 17. Let $\mathcal{P} = \{p \vee q \leftarrow;\ \ r \vee s \leftarrow \neg q\}$. We calculate $\mathsf{KK}(\mathcal{IC}_{\mathcal{P}})$ as follows:

- $\mathcal{IC}'_{\mathcal{P}}(\emptyset, \{p,q,r,s\}) = (\{\{p\},\{q\}\}\uparrow, \{\{p,q,r,s\}\}\downarrow)$. We obtain this by observing that $\mathcal{IC}^l_{\mathcal{P}}(\emptyset, \{p,q,r,s\}) = \{\{p\},\{q\},\{p,q\}\}$ and $\mathcal{IC}^u_{\mathcal{P}}(\emptyset, \{p,q,r,s\}) = \{\{p,r\},\{p,s\},\{q,r\}\{q,r\},\{p,q,r\},\{p,q,s\},\{p,r,s\},\{q,r,s\},\{p,q,r,s\}\}\}$.
- $\mathcal{IC}'_{\mathcal{P}}((\{\{p\},\{q\}\}\uparrow, \{\{p,q,r,s\}\}\downarrow)) = (\{\{p\},\{q\}\}\uparrow, \{\{p,q,r,s\}\}\downarrow)$ and thus a fixpoint is reached.
(It should be noticed that in the second step of the iteration, two more precise upper bounds are obtained as well: in view of $(\{p\},\{p,r\})(\neg q) = (\{p\},\{p,s\})(\neg q) = \mathsf{T}$, $\mathcal{IC}_{\mathcal{P}}(\{p\},\{p,r\})\uparrow = \mathcal{IC}_{\mathcal{P}}(\{p\},\{p,s\})\uparrow = \{\{p,r\},\{p,s\},\{q,r\},\{q,s\}\}\uparrow$, but this lower bound becomes "nulified" since e.g. $\mathcal{IC}_{\mathcal{P}}(\{q\},\{q,r\})\uparrow$ contains $\{p\}\uparrow$.)

The Kripke-Kleene fixpoint that is reached is represented by the convex set

$$\{\{p\},\{q\},\{p,r\},\{p,s\},\{q,r\},\{q,s\},\{p,q,r\},\{p,q,s\},\{p,r,s\},\{q,r,s\},\{p,q,r,s\}\}.$$

Intuitively, the meaning of this Kripke-Kleene state is that every two-valued (stable) model of this program (if they exist), is in this convex set. This is indeed the case, as the stable models of this program are $\{p,r\}$, $\{p,s\}$ and $\{q\}$.

This example thus illustrates how the Kripke-Kleene state is an approximation of the semantics of disjunctive logic programs. The main benefit of this is that it is guaranteed to uniquely exist and be constructively computed.

To summarize the results in this section, we have shown the following: There are two ways to generalize the Kripke-Kleene semantics from deterministic AFT to a non-deterministic setting: one can keep the type the same, resulting in *Kripke-Kleene interpretations* or simply fixpoints of an ndao. We have shown that such fixpoints of $\mathcal{IC}_{\mathcal{P}}$ correspond to weakly supported models. Uniqueness and existence are not guaranteed. This is solved by the alternative generalization of Kripke-Kleene semantics from deterministic AFT, by *Kripke-Kleene state*, which is a set of interpretations instead of a single interpretation. Existence, uniqueness, and consistency for exact ndso's are guaranteed for these states, and, moreover, for deterministic operators the two Kripke-Kleene fixpoints coincide.

5. Stable semantics

In this section, the stable semantics from deterministic AFT is generalized to the non-deterministic setting. In Section 5.1, we first introduce and study the stable operator and the corresponding stable interpretation semantics. Then, in Section 5.2 we study the well-founded state semantics. In Section 5.3, we show the usefulness of the well-founded state semantics by relating it to the well-founded semantics with disjunction [2] for disjunctive logic programming.

5.1. Stable interpretation semantics

In deterministic AFT, the idea behind the stable operator is to find fixpoints that are minimal w.r.t. the truth order by constructing a new lower bound and a new upper bound on the basis of the current upper respectively lower bound. In more detail, given the current upper bound y, we look for the $\leq$-least fixpoint of $\mathcal{O}_l(\cdot, y)$ (which is guaranteed to exist for a deterministic approximation fixpoint operator $\mathcal{O}$ and coincides with the greatest lower bound of fixpoints of $\mathcal{O}_l(\cdot, y)$). Thus, informally, we look for the smallest lower bound s.t. the operator $\mathcal{O}$ lets us derive nothing more and nothing less when assuming y as an upper bound.

Instead of generating a single lower and a single upper bound, an ndao generates a set of lower bounds $\{x_1, x_2, \ldots\}$ and a set of upper bounds $\{y_1, y_2, \ldots\}$ on the basis of a lower and upper bound (x, y). We can again look for a (not necessarily unique) smallest lower bound that an ndao allows us to derive in view of a given upper bound y by looking for $\leq$-minimal fixpoints of $\mathcal{O}_l(\cdot, y)$. We will see below that only under certain assumptions on the ndao such a fixpoint is guaranteed to exist. Furthermore, other properties of the stable operator and fixpoints, such as $\preceq_i^A$-monotonicity, the existence of stable fixpoints and the $\leq_t$-minimality of stable fixpoints, do not generalize or only generalize under certain assumptions from the deterministic setting. Nevertheless, the fixpoint and state semantics based on this construction are useful in general, as they can e.g. characterize the stable semantics for disjunctive logic programs and the *well-founded semantics with disjunction* by Alcântara, Damásio and Pereira [2], and for positive programs it coincides with the set of minimal models (Proposition 15).

This section is organized as follows. We first define stable non-deterministic operators. Then, we study their properties, mainly by looking at how properties from the deterministic setting can be generalized. We first show that non-deterministic stable operators and their fixpoints faithfully generalize the corresponding deterministic notions (see Proposition 11 and Corollary 3). Next, we study conditions under which the stable operator is well-defined (culminating in Proposition 13). Thereafter, we show that stable operators are not guaranteed to be $\preceq_i^A$-monotonic (Example 22), and that stable fixpoints are not guaranteed to exist (Example 22). Finally, we show that, under certain conditions, stable fixpoints are $\leq_t$-minimal fixpoints of an ndao (Proposition 14). This general study of stable operators and their fixpoints is followed by an illustration of their usefulness, where it is shown that (partial) stable interpretations of disjunctive logic programs are stable fixpoints of $\mathcal{IC}_{\mathcal{P}}$ (Theorem 4).

Definition 20. Let $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L}) \times \wp(\mathcal{L})$ be an ndao. We define:

- The *complete lower stable operator*: (for any $y \in \mathcal{L}$)

$$C(\mathcal{O}_l)(y) \ = \ \{x \in \mathcal{L} \mid x \in \mathcal{O}_l(x, y) \text{ and } \neg\exists x' < x : x' \in \mathcal{O}_l(x', y)\}.$$

- The *complete upper stable operator*: (for any $x \in \mathcal{L}$)

$$C(\mathcal{O}_u)(x) \;=\; \{y \in \mathcal{L} \mid y \in \mathcal{O}_u(x,y) \text{ and } \neg\exists y' < y : y' \in \mathcal{O}_u(x,y')\}.$$

- The *stable operator*: $S(\mathcal{O})(x,y) = (C(\mathcal{O}_l)(y),\, C(\mathcal{O}_u)(x))$
- A *stable fixpoint* of $\mathcal{O}$ is any $(x,y) \in \mathcal{L}^2$ such that $(x,y) \in S(\mathcal{O})(x,y)$.[21]

Example 18. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow\}$ from Example 1. It holds that for any $x, y \subseteq \{p,q\}$, $\mathcal{IC}^l_{\mathcal{P}}(x,y) = \{\{p\},\{q\},\{p,q\}\}$ and thus $C(\mathcal{IC}^l_{\mathcal{P}}(y)) = \{\{p\},\{q\}\}$. It thus follows that $(\{p\},\{p\})$ and $(\{q\},\{q\})$ are the stable fixpoints of $\mathcal{IC}_{\mathcal{P}}$.

The complete operator based on $\mathcal{IC}_{\mathcal{P}}$ produces the minimal models of the reducts w.r.t. the input. Before showing this, we first recall that the *two-valued models* of a positive program $\mathcal{P}$ are the sets $x \subseteq \mathcal{A}_{\mathcal{P}}$ s.t. for every $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$, $(x,x)(\phi) = \mathsf{T}$ implies $x \cap \Delta \neq \emptyset$ (Footnote 6). In what follows, we denote the set of two-valued models $\mathcal{P}$ by $mod_2(\mathcal{P})$.

Proposition 10. *Consider a dlp $\mathcal{P}$ and some $y \subseteq \mathcal{A}_{\mathcal{P}}$. Then $C(\mathcal{IC}^l_{\mathcal{P}})(y) = \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{y}))$.*[22]

Remark 13. Notice that we did *not* define the complete stable operator as the glb of fixpoints of $\mathcal{O}_l(\cdot,x)$. Indeed, this leads to several problems. First, the glb of fixpoints of $\mathcal{O}_l(\cdot,x)$ might itself not be a fixpoint of $\mathcal{O}_l(\cdot,x)$ (since we cannot apply Tarski-Knaster fixpoint theorem to the operator $\mathcal{O}_l(\cdot,x) : \mathcal{L} \to \wp(\mathcal{L})$). Secondly, and more importantly, taking the glb might lead to a loss of information. Consider e.g. the program $\{p \vee q \leftarrow\}$. Then $\mathcal{IC}^l_{\mathcal{P}}(\cdot,x)$ has three fixpoints (for any $x \subseteq \{p,q\}$), namely $\{p\}$, $\{q\}$ and $\{p,q\}$. If we take the glb, however, we obtain $\emptyset$. This would be counter-intuitive, since we clearly should be more interested in the more informative $\{p\}$ and $\{q\}$, which represent two possible choices to be made in view of $p \vee q \leftarrow$.

Remark 14. Notice that since $\mathcal{O}_l(\cdot,x)$ maps from $\mathcal{L}$ to $\wp(\mathcal{L})$, i.e., it is not a deterministic operator on $\mathcal{L}$, we cannot use Tarski-Knaster fixpoint theorem to guarantee the existence of fixpoint of $\mathcal{O}_l(\cdot,x)$, i.e., $C(\mathcal{O}_l)$ is not guaranteed to be non-empty (i.e., there might be some $y \in \mathcal{L}$ such that $C(\mathcal{O}_l)(y) = \emptyset$). Indeed, as a case in point, consider the following example:

Example 19. Consider the lattice $L = \langle \mathbb{N}^- \cup \{-\infty\}, \leq \rangle$ where $\mathbb{N}^- = \{0,-1,-2,\ldots\}$ and $\leq$ is defined as usual (e.g. $-2 \leq -1$). Consider the operator $\mathcal{O}_l(x,y)$ defined by $\mathcal{O}_l(x,y) = \{x\}$ for any $x,y \in \mathbb{N}^-$, and $\mathcal{O}_l(-\infty,y) = \mathbb{N}^-$ for any $y \in \mathbb{N}^-$. Notice that for this example, the value of $\mathcal{O}_l(x,-\infty)$ is not significant. It can be verified that this operator is $\preceq^S_L$-monotonic. Indeed, if $x_1,x_2,y_1,y_2 \in \mathbb{N}^-$, then if $(x_1,y_1) \leq (x_2,y_2)$, $x_1 \leq x_2$ and $\mathcal{O}_l(x_1,y_1) = \{x_1\} \preceq^S_L \mathcal{O}_l(x_2,y_2) = \{x_2\}$ since $x_1 \leq x_2$. Furthermore, if $x_1 = -\infty$ then for every x_2 s.t. $(x_1,y_1) \leq (x_2,y_2)$, $x_2 \in \mathcal{O}_l(x_1,y_1) = \mathbb{N}^-$ and thus $\mathcal{O}_l(x_1,y_1) = \mathbb{N}^- \preceq^S_L \mathcal{O}_l(x_2,y_2)$. Moreover, for any $y \in \mathbb{N}^-$, every $x \in \mathbb{N}^-$ is a fixpoint of $\mathcal{O}_l(\cdot,y)$ (as $\{x\} = \mathcal{O}_l(x,y)$). Since $\mathbb{N}^-$ has no $\leq$-minimal element, $C(\mathcal{O}_l)(y) = \emptyset$ for every $y \in \mathbb{N}^-$.

In what follows (see Definition 22), we will delineate a condition that guarantees that the complete lower and upper stable operator is non-empty.

Next, we show that in *finite* bilattices, the definitions of complete stable operators coincide with those for deterministic AFT. This holds since in finite lattices, the greatest lower bound coincides with the minimum.

Proposition 11. *Let $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ be an ndao over a finite, complete lattice $\mathcal{L}$, where $\mathcal{O}(x,y)$ is a pair of singleton sets for every $x,y \in \mathcal{L}$. Let $\mathcal{O}^{\mathsf{AFT}}$ be the deterministic approximation operator defined by $\mathcal{O}^{\mathsf{AFT}}(x,y) = (w,z)$ where $\mathcal{O}(x,y) = (\{w\},\{z\})$.*[23] *Then $C(\mathcal{O}_l)(y) = \{C(\mathcal{O}^{\mathsf{AFT}}_l)(y)\}$, and $C(\mathcal{O}_u)(x) = \{C(\mathcal{O}^{\mathsf{AFT}}_u)(x)\}$ for every $x,y \in \mathcal{L}$.*[24]

Proof. By Proposition 3 $\mathcal{O}^{\mathsf{AFT}}$ is a deterministic approximation operator. The equalities in the proposition are immediate since for finite lattices, $\{x \in \mathcal{L} \mid x \in \mathcal{O}_l(x,y) \text{ and } \neg\exists x' < x : x' \in \mathcal{O}_l(x',y)\} = \{\bigsqcap\{x \in \mathcal{L} \mid x \in \mathcal{O}_l(x,y)\}\}$. □

Thus, in view of Proposition 11, for finite lattices the notion of stable fixpoint is not changed in the non-deterministic case:

Corollary 3. *Let $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ be an ndao over a finite, complete lattice $\mathcal{L}$, where $\mathcal{O}(x,y)$ is a pair of singleton sets for every $x,y \in \mathcal{L}$. Let $\mathcal{O}^{\mathsf{AFT}}$ be defined by $\mathcal{O}^{\mathsf{AFT}}(x,y) = (w,z)$ where $\mathcal{O}(x,y) = (\{w\},\{z\})$. Then (x,y) is a stable fixpoint of $\mathcal{O}$ according to Definition 20 iff (x,y) is a stable fixpoint of $\mathcal{O}^{\mathsf{AFT}}$ according to Definition 6.*

[21] Notice that we slightly abuse notation and write $(x,y) \in S(\mathcal{O})(x,y)$ to abbreviate $x \in (S(\mathcal{O})(x,y))_1$ and $y \in (S(\mathcal{O})(x,y))_2$, i.e. x is a lower bound generated by $S(\mathcal{O})(x,y)$ and y is an upper bound generated by $S(\mathcal{O})(x,y)$.

[22] Recall Definition 3.

[23] Recall also Proposition 3.

[24] Notice that since $\mathcal{O}^{\mathsf{AFT}}$ is a deterministic approximation operator, $C(\mathcal{O}^{\mathsf{AFT}}_l)(y)$ and $C(\mathcal{O}^{\mathsf{AFT}}_u)(y)$ are taken as in Definition 6.

Remark 15. For approximation operators over infinite bilattices, the coincidence in Proposition 11 cannot be guaranteed. The reason is that the minimum taken in the non-deterministic complete operators $C(\mathcal{O}^u)$ (and $C(\mathcal{O}^l)$) might not coincide with the glb taken in the deterministic complete operators $C(\mathcal{O}_l^{\mathsf{AFT}})(y)$ and $C(\mathcal{O}_u^{\mathsf{AFT}})(y)$. A case in point is the following example:

Example 20. Consider a lattice $L = \langle \{\bot, \top\} \cup \{x_i \mid i \in \mathbb{N}\}, \leq \rangle$ where $\bot < x_i < \top$ for every $i \in \mathbb{N}$. Consider the operator $\mathcal{O}$ defined as follows:

- $\mathcal{O}_l(x_i, y) = \mathcal{O}_u(y, x_i) = \{x_i\}$ for any $y \in \{\bot, \top\} \cup \{x_i \mid i \in \mathbb{N}\}$ and $i \in \mathbb{N}$,
- $\mathcal{O}_l(\bot, x_i) = \mathcal{O}_u(x_i, \bot) = \{x_i\}$ for any $i \in \mathbb{N}$,
- $\mathcal{O}_l(x, \top) = \mathcal{O}_u(\top, x) = \{\top\}$ for any $x \in \{\top\} \cup \{x_i \mid i \in \mathbb{N}\}$, and
- $\mathcal{O}_l(\bot, \top) = \{\bot\}$.

Then $C(\mathcal{O}_l)(x_i) = \{x_i \mid i \in \mathbb{N}\}$, whereas $C(\mathcal{O}_l^{\mathsf{AFT}})(x_i) = \bigsqcap\{x_i \mid i \in \mathbb{N}\} = \bot$ for any $i \in \mathbb{N}$.

The next property we study is the well-definedness of the complete (and thus stable) operator, which is not guaranteed in view of Remark 14. First, we show two useful lemmas (the first one is based on a similar lemma in the paper by Pelov and Truszczyński [48]), which show that minimal pre-fixpoints are also minimal fixpoints and vice versa. For this, we first generalize the notion of a pre-fixpoint from the deterministic setting to non-deterministic operators.

Definition 21. We say that $w \in \mathcal{L}$ is a *pre-fixpoint* of the non-deterministic operator O on $\mathcal{L}$, if $O(w) \preceq_L^S \{w\}$.

It can be easily verified that this generalizes the notion of a pre-fixpoint of a deterministic operator.

Lemma 10. *Let $O : \mathcal{L} \rightarrow \wp(\mathcal{L})$ be a $\preceq_L^S$-monotonic non-deterministic operator. Then if w is a $\leq$-minimal pre-fixpoint of O, it is a $\leq$-minimal fixpoint of O.*[25]

Proof. Suppose that w is a $\leq$-minimal pre-fixpoint of O, i.e., $O(w) \preceq_L^S \{w\}$. This means that there is some $z \in O(w)$ s.t. $z \leq w$. Since O is $\preceq_L^S$-monotonic, $O(z) \preceq_L^S O(w)$. Since $z \in O(w)$, $O(z) \preceq_L^S \{z\}$, i.e. z is a pre-fixpoint of O. Since w is a $\leq$-minimal pre-fixpoint and $z \leq w$, $z = w$. Minimality is immediate since fixpoints are in particular pre-fixpoints. □

Lemma 11. *Let $O : \mathcal{L} \rightarrow \wp(\mathcal{L})$ be a $\preceq_L^S$-monotonic non-deterministic operator. Then if w is a $\leq$-minimal fixpoint of O, it is a $\leq$-minimal pre-fixpoint of O.*

Proof. Suppose that w is a $\leq$-minimal fixpoint of O. Then $w \in O(w)$ and so $O(w) \preceq_L^S \{w\}$. Thus, w is a pre-fixpoint. Suppose now towards a contradiction that for some $w' < w$, $O(w') \preceq_L^S \{w'\}$. Without loss of generality, we may assume that w' is a minimal pre-fixpoint. But then, by Lemma 10, it is a minimal fixpoint of O, a contradiction. □

Uniqueness of this $\leq$-minimal fixpoint cannot be guaranteed (as can be seen in e.g. Example 11). This is a crucial difference with deterministic operators (where even a unique $\leq$-*least* fixpoint is guaranteed to exist). Thus, to summarize, non-determinism forces us to take $\leq$-minimal fixpoints instead of the greatest lower bound. This choice, in turn, means that existence is not guaranteed on infinite lattices. Therefore, we now turn to conditions that ensure the existence of $\leq$-minimal fixpoint of non-deterministic operators over infinite lattices. As we shall see (Proposition 13 below), the next property (inspired by Pelov and Truszczyński [48]) assures existence. In order to disambiguate the order over ordinals from the lattice order, we will denote the former by $\prec$.

Definition 22. A non-deterministic operator $O : \mathcal{L} \rightarrow \wp(\mathcal{L})$ is *downward closed* if for every sequence $X = \{x_\epsilon\}_{\epsilon \prec \alpha}$ of elements in $\mathcal{L}$ such that:

1. for every $\epsilon \prec \alpha$, it holds that $O(x_\epsilon) \preceq_L^S \{x_\epsilon\}$, and
2. for every $\epsilon' \prec \epsilon \prec \alpha$, it holds that $x_{\epsilon'} < x_\epsilon$,

we have that $O(\bigsqcap X) \preceq_L^S \bigsqcap(X)$.

Definition 22 is a generalization of a similar definition by Pelov and Truszczyński [48]. Its says that an operator is downward closed if the greatest lower bound of every chain of pre-fixpoints is itself a pre-fixpoint. As we will see in Proposition 14, this ensures that O admits a fixpoint.

As an example of an operator that is downward closed, we consider $\mathcal{IC}_{\mathcal{P}}^l(\cdot, y)$ (recall Definition 12).[26]

[25] Notice that for any ndao $\mathcal{O}$, $\mathcal{O}_l(\cdot, x)$ is an operator of the type $\mathcal{O} : \mathcal{L} \rightarrow \wp(\mathcal{L})$ and thus a non-deterministic operator, which we denote by O.

[26] This is inspired by the proof of a similar result by Pelov and Truszczyński [48].

Proposition 12. *For any dlp $\mathcal{P}$ and any $y \subseteq \mathcal{A}_\mathcal{P}$, $\mathcal{IC}_\mathcal{P}(\cdot, y)$ is downward closed.*

Next we show that downward closure is indeed a sufficient condition for assuring the existence of a fixpoint:

Proposition 13. *Let $\mathcal{L} = \langle L, \leq \rangle$ be a lattice and let $O : \mathcal{L} \to \wp(\mathcal{L})$ be a downward closed, $\preceq_L^S$-monotonic non-deterministic operator. Then O admits a $\leq$-minimal fixpoint.*

Proof. By Lemma 10 it is sufficient to show that O admits a $\leq$-minimal pre-fixpoint. The set of pre-fixpoints of O is clearly a partially ordered set. With downwards closedness, every chain of pre-fixpoints has a lower bound (which is also a pre-fixpoint). Thus, by the Kuratowski-Zorn lemma [41,62], the set of pre-fixpoints has a minimal element. □

Notice that Proposition 13 also ensures the well-definedness of $C(\mathcal{O}_u)$ if $\mathcal{O}$ is symmetric. The investigation of other conditions of well-definedness of the stable operator is left for future work.

We obtain the following corollary for the approximation operator $\mathcal{IC}^l_\mathcal{P}$:

Corollary 4. *For every dlp $\mathcal{P}$, $\mathcal{IC}^l_\mathcal{P}(\cdot, y)$ has a $\leq$-minimal fixpoint.*

Proof. By Propositions 12 and 13, and since $\mathcal{IC}^l_\mathcal{P}(\cdot, y)$ is $\preceq_L^S$-monotonic. □

As we have now established conditions under which the stable operator is well-defined, we turn to the property of $\preceq_i^A$-monotonicity of the stable operator. We notice that in general, $S(\mathcal{O})$ is *not* a $\preceq_i^A$-monotonic operator:

Example 21. Consider the program $\mathcal{P} = \{p \vee q \leftarrow; \quad p \leftarrow \neg r\}$. We calculate the applications of the stable operators as follows:

- Since $(\{p\}, \{r\})(\neg r) = (\{q\}, \{r\})(\neg r) = \mathsf{F}$, it holds that $C(\mathcal{IC}^l_\mathcal{P})(\{r\}) = \{\{p\}, \{q\}\}$ for any $x \subseteq \mathcal{A}_\mathcal{P} \setminus \{r\}$.
- Since $(\{r\}, x)(\neg r) = \mathsf{F}$ for any $x \subseteq \mathcal{A}_\mathcal{P} \setminus \{r\}$, it holds that $\mathcal{IC}^u_\mathcal{P}(x, \{r\}) = \mathcal{IC}^l_\mathcal{P}(\{r\}, x) = \{\{p\}, \{q\}\}$ and thus $C(\mathcal{IC}^u_\mathcal{P}(\{r\}) = \{\{p\}, \{q\}\}$.
- Since $(\emptyset, x)(\neg r) = \mathsf{T}$ for any $x \subseteq \mathcal{A}_\mathcal{P} \setminus \{r\}$, it holds that $C(\mathcal{IC}^u_\mathcal{P})(\{r\}) = \{\{p\}\}$.

Altogether, this means that

$$S(\mathcal{IC}_\mathcal{P})(\emptyset, \{r\})) = (\{\{p\}, \{q\}\}, \{\{p\}\}) \not\preceq_i^A S(\mathcal{IC}_\mathcal{P})(\{r\}, \{r\}) = (\{\{p\}, \{q\}\}, \{\{p\}, \{q\}\})$$

$$\text{since } \{\{p\}, \{q\}\} \not\preceq_L^H \{\{p\}\} \text{ as } \{q\} \not\subseteq \{p\}$$

And thus, even though $(\emptyset, \{r\})) \leq_i (\{r\}, \{r\})$, it does *not* hold that $S(\mathcal{IC}_\mathcal{P})(\emptyset, \{r\})) \preceq_i^A S(\mathcal{IC}_\mathcal{P})(\{r\}, \{r\})$.

Recall that (in contrast to the last example), by Proposition 7, the state version of the stable operator *is* still $\preceq_i^A$-monotonic. Thus, we can still construct a state by iteratively applying the state version of the stable operator. We will detail this construction in Section 5.2 and show that this state, which we call the *well-founded state*, exists, is unique, is more precise than the Kripke-Kleene state, and coincides with the well-founded fixpoint for deterministic operators.

A third property we investigate is the existence of stable fixpoints.[27] Even when the complete stable operator is non-empty for every element of a lattice, stable fixpoints may not exist:

Example 22. Consider the following dlp: $\mathcal{P} = \{p \vee q \vee r \leftarrow; \quad p \leftarrow \neg q; \quad r \leftarrow \neg p; \quad q \leftarrow \neg r\}$. It can be checked that there are no $x, y \subseteq \{p, q, r\}$ s.t. $(x, y) \in S(\mathcal{IC}_\mathcal{P})(x, y)$. To make this clearer, we calculate some of the outcomes of the complete stable operator $C(\mathcal{IC}^l_\mathcal{P})$:

- $C(\mathcal{IC}^l_\mathcal{P})(\emptyset) = C(\mathcal{IC}^u_\mathcal{P})(\emptyset) = \{\{p, q, r\}\}$.
- $C(\mathcal{IC}^l_\mathcal{P})(\{p\}) = C(\mathcal{IC}^u_\mathcal{P})(\{p\}) = \{\{p, q\}\}$.
- $C(\mathcal{IC}^l_\mathcal{P})(\{p, q\}) = C(\mathcal{IC}^u_\mathcal{P})(\{p\}) = \{\{q\}\}$.
- $C(\mathcal{IC}^l_\mathcal{P})(\{p, q, r\}) = C(\mathcal{IC}^u_\mathcal{P})(\{p\}) = \{\{p\}, \{q\}, \{r\}\}$.

Notice that $C(\mathcal{IC}^u_\mathcal{P}(x)) = C(\mathcal{IC}^l_\mathcal{P}(x))$ for any $x \subseteq \mathcal{A}_\mathcal{P}$ as $\mathcal{IC}_\mathcal{P}$ is symmetric. Other cases can be easily derived in view of symmetry of $\mathcal{IC}^l_\mathcal{P}$ w.r.t. p, q and r. As there is no $x, y \subseteq \mathcal{A}_\mathcal{P}$ for which $x \in C(\mathcal{IC}^l_\mathcal{P})(y)$ and $y \in C(\mathcal{IC}^u_\mathcal{P})(x) = C(\mathcal{IC}^l_\mathcal{P})(x)$, we conclude that

[27] Notice that this is not the same as the existence of $\leq$-minimal fixpoints of $\mathcal{O}^l(\cdot, y)$: the latter establishes merely that $S(\mathcal{O})$ is well-defined, but does not guarantee that stable fixpoints exist as we will see now.

no stable interpretation exists, which is in accordance with the stable model semantics for disjunctive logic programming, where a well-founded or more generally three-valued stable model does not exist in this case.

The next proposition shows that if stable fixpoints of $\mathcal{O}$ exist, they are $\leq_t$-minimal fixpoints of $\mathcal{O}$, thus generalizing this property from the deterministic to the non-deterministic case.

Proposition 14. *Let $L = \langle \mathcal{L}, \leq \rangle$ be a lattice and let $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L}) \times \wp(\mathcal{L})$ be an ndao. Then every stable fixpoint of $\mathcal{O}$ is a $\leq_t$-minimal fixpoint of $\mathcal{O}$.*

Proof. We first show that any stable fixpoint of $\mathcal{O}$ is a fixpoint of $\mathcal{O}$. Suppose that $(x, y) \in S(\mathcal{O})(x, y)$. Then $x \in C(\mathcal{O}_l)(y)$ and $y \in C(\mathcal{O}_u)(x)$, which implies that $x \in \mathcal{O}_l(x, y)$ and $y \in \mathcal{O}_u(x, y)$. Thus, $(x, y) \in \mathcal{O}(x, y)$, i.e. (x, y) is a fixpoint of $\mathcal{O}$.

We now show that any stable fixpoint of $\mathcal{O}$ is a $\leq_t$-minimal fixpoint of $\mathcal{O}$. Consider some $(x', y') \leq_t (x, y)$ with $(x', y') \in \mathcal{O}(x', y')$. Since $\mathcal{O}_l(x', \cdot)$ is $\preceq^S_L$-anti-monotonic (Lemma 2), $\mathcal{O}_l(x', y) \preceq^S_L \mathcal{O}_l(x', y')$, which means, as $x' \in \mathcal{O}_l(x', y')$, that $\mathcal{O}_l(x', y) \preceq^S_L \{x'\}$. Thus, x' is a pre-fixpoint of $\mathcal{O}_l(\cdot, y)$. By Lemma 11, x is a minimal pre-fixpoint of $\mathcal{O}_l(\cdot, y)$. Since $x' \leq x$, necessarily, $x' = x$. This means that, as $y' \in \mathcal{O}_u(x, y')$, y' is a fixpoint of $\mathcal{O}_u(x, \cdot)$. As $y \in C(\mathcal{O}_u)(x)$, y is a $\leq$-minimal fixpoint of $\mathcal{O}_u(x, \cdot)$. As $y' \leq y$, this means that $y = y'$. $\square$

Even though they are not guaranteed to exist, stable fixpoints are useful in knowledge representation. For example, for any DLP $\mathcal{P}$, the stable fixpoints of $\mathcal{IC}_\mathcal{P}$ characterize the (three-valued) stable models of $\mathcal{P}$:

Theorem 4. *Consider a normal disjunctive logic program $\mathcal{P}$ and a consistent interpretation $(x, y) \in \wp(\mathcal{A}_\mathcal{P}) \times \wp(\mathcal{A}_\mathcal{P})$. Then (x, y) is a stable model of $\mathcal{P}$ iff $(x, y) \in S(\mathcal{IC}_\mathcal{P})(x, y)$.*

Notice that this also means that two-valued stable models coincide with total stable fixpoints (i.e., x is a two-valued stable model iff $(x, x) \in S(\mathcal{IC}_\mathcal{P})(x, x)$).

Example 23. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow\}$ from Example 1 (see also Example 18). In view of Theorem 4, it is not a coincidence that $(\{p\}, \{p\})$ and $(\{q\}, \{q\})$ are the stable fixpoints of $\mathcal{IC}_\mathcal{P}$ and the stable interpretations of $\mathcal{P}$.

5.2. Well-founded state semantics

The well-founded fixpoint in determinstic AFT is obtained by iteratively applying the stable operator $S(\mathcal{O})$ to the least precise pair $(\bot, \top)$, which results in a fixpoint that approximates any fixpoint of the operator O approximated by $\mathcal{O}$ and is guaranteed to exist, be unique, and is at least as precise as the Kripke-Kleene fixpoint. In this section, we generalize this construction to the non-deterministic setting by defining the *well-founded state*, a convex set that is unique, guaranteed to exist, at least as percise as the Kripke-Kleene state, approximates any fixpoint of the non-deterministic operator O approximated by $\mathcal{O}$, and is obtained by iteratively applying $S(\mathcal{O})$. Thus, the well-founded state $\mathsf{WF}(\mathcal{O})$ is obtained, for an approximation operator $\mathcal{O}$, by first taking the stable operator $S(\mathcal{O})$ on the basis of $\mathcal{O}$ and then taking the Kripke-Kleene state $\mathsf{KK}(S(\mathcal{O}))$ of this operator (i.e., iterating the application of the state-version of the operator starting from the least precise element until a fixpoint is reached), thus resulting in $\mathsf{WF}(\mathcal{O}) = \mathsf{KK}(S(\mathcal{O}))$.

For clarity, we deduce a somewhat simpler computation of the state operator $(S(\mathcal{O}))'(X, Y)$:

$$\Big(\Big(\bigcup_{y\in Y} C(\mathcal{O}_l)(y)\Big)\uparrow, \Big(\bigcup_{x\in X} C(\mathcal{O}_u)(x)\Big)\downarrow\Big).$$

Notice that the calculation of new lower bounds depends only on the old upper bounds, and the calculation of the new upper bounds depends only on the old lower bounds.

We first prove the following useful lemma, which shows that, when building up the well-founded state, applications of $S(\mathcal{O})'$ result in at least as precise convex sets as applications of $\mathcal{O}$.

Lemma 12. *Let a lattice $\mathcal{L} = \langle L, \leq \rangle$ and an ndao $\mathcal{O}$ over $\mathcal{L}$ be given s.t. $\mathcal{O}_l(\cdot, x)$ is downward closed for every $x \in L$. Then for any ordinal α, $(\mathcal{O}')^\alpha(\bot, \top) \preceq^A_i ((S(\mathcal{O}))')^\alpha(\bot, \top)$.*

Proof. We show this by induction on α.

For the base case, we show that $\mathcal{O}(\bot, \top) \preceq^A_i S(\mathcal{O})(\bot, \top)$. Let $S(\mathcal{O})(\bot, \top) = (X, Y)$. Consider some $z \in C(\mathcal{O}_l)(\top)$. I.e. $z \in \mathcal{O}_l(z, \top)$ and for every $w \in \mathcal{O}_l(w, \top)$, $w \not< z$. Since $(\bot, \top) \leq_i (z, \top)$, with the $\preceq^A_i$-monotonicity of $\mathcal{O}$, $\mathcal{O}_l(\bot, \top) \preceq^S_L \mathcal{O}_l(z, \top)$. Since this argument holds for an arbitrary $z \in C(\mathcal{O}_l)(\top)$, we have established $\mathcal{O}_l(\bot, \top) \preceq^S_L C(\mathcal{O})(\top)$. The proof for $C(\mathcal{O}_u)(\bot)$ (i.e., that $C(\mathcal{O}_u)(\bot)) \preceq^H_L \mathcal{O}_u(\bot, \top)$ is similar.

For the inductive case, consider two ordinals α and β, let β be the successor ordinal of α, and assume that $(\mathcal{O}')^\alpha(\bot, \top) \preceq^A_i ((S(\mathcal{O}))')^\alpha(\bot, \top)$. Let $((S(\mathcal{O}))')^\alpha(\bot, \top) = (X, Y)$ and consider some $y \in Y$. Let $x \in X$. Recall that $x \in C(\mathcal{O}_l)(y)$ means that x is a $\leq$-minimal fixpoint of $\mathcal{O}_l(\cdot, y)$, and thus $x \in \mathcal{O}_l(x, y)$. Notice that $(\bot, \top) \leq_i (x, y)$ and so $\mathcal{O}(\bot, \top) \preceq^A_i \mathcal{O}(x, y)$, which implies that

$\mathcal{O}_l(\bot,\top) \preceq^S_L \mathcal{O}_l(x,y)$. This means that $\mathcal{O}_l(\bot,\top) \preceq^S_L \{x\}$ (since $x \in \mathcal{O}_l(x,y)$). By the inductive hypothesis, $\{y\} \preceq^H_L \mathcal{O}^\alpha_l(\bot,\top)$, which means (for any ordinal γ smaller than α), since $\mathcal{O}'$ is $\preceq^A_i$-monotonic and $\mathcal{O}^\alpha_u(\bot,\top) \preceq^H_L \mathcal{O}^\gamma_u(\bot,\top)$, that $\{y\} \preceq^H_L \mathcal{O}^\gamma_l(\bot,\top)$. Thus, $\mathcal{O}(\bot,\top) \preceq^A_i (x,y)$. We can now use the same line of reasoning recursively until we reach the ordinal α to obtain $(\mathcal{O}')^\alpha(\bot,\top) \preceq^A_i (x,y)$. Applying $\mathcal{O}'$ one more time gives us $(\mathcal{O}')^\beta(\bot,\top) \preceq^A_i \mathcal{O}(x,y)$, which, with $x \in \mathcal{O}(x,y)$ means $(\mathcal{O}'_l)^\beta(\bot,\top) \preceq^S_L \{x\}$, as desired. The proof that $\{y\} \preceq^H_L (\mathcal{O}'_u)^\beta(\bot,\top)$ for every $y \in Y$ is similar.

For a limit ordinal α, we have to show that $lub_{\preceq^A_i}\{(\mathcal{O}')^\beta(\bot,\top) \mid \beta < \alpha\} \preceq^A_i lub_{\preceq^A_i}\{(\mathcal{S}(\mathcal{O})')^\beta(\bot,\top) \mid \beta < \alpha\}$ under the assumption that $(\mathcal{O}')^\beta(\bot,\top) \preceq^A_i ((\mathcal{S}(\mathcal{O}))')^\beta(\bot,\top)$ for any $\beta < \alpha$. This is immediate, in view of the following considerations:

1. By Corollary 1, $lub_{\preceq^A_i}\{(\mathcal{O}')^\beta(\bot,\top) \mid \beta < \alpha\} = (\bigcap_{\beta<\alpha}((\mathcal{O}')^\beta(\bot,\top))_1, \bigcap_{\beta<\alpha}((\mathcal{O}')^\beta(\bot,\top))_2)$, and $lub_{\preceq^A_i}\{(\mathcal{S}(\mathcal{O})')^\beta(\bot,\top) \mid \beta < \alpha\} = (\bigcap_{\beta<\alpha}((\mathcal{S}(\mathcal{O}'))^\beta(\bot,\top))_1, \bigcap_{\beta<\alpha}((\mathcal{S}(\mathcal{O}'))^\beta(\bot,\top))_2)$.
2. By the inductive hypothesis, it holds that $\bigcap_{\beta<\alpha}((\mathcal{O}')^\beta(\bot,\top))_1 \preceq^S_L \bigcap_{\beta<\alpha}((\mathcal{S}(\mathcal{O}'))^\beta(\bot,\top))_1$ and $((\mathcal{S}(\mathcal{O}'))^\beta(\bot,\top))_2 \preceq^H_L \bigcap_{\beta<\alpha}((\mathcal{O}')^\beta(\bot,\top))_2$. □

Using the above lemma, we can now show that, for downwards closed operators $\mathcal{O}$, the well-founded state $\mathsf{WF}(\mathcal{O})$, which we define as the Kripke-Kleene-state of the stable operator $\mathsf{KK}(\mathcal{S}(\mathcal{O}))$, is at least as precise as the Kripke-Kleene-state of $\mathcal{O}$, approximates the stable fixpoints of $\mathcal{O}$ and the fixpoints of the approximated operator O:

Theorem 5. *Let $\mathcal{L} = \langle L, \leq\rangle$ be a lattice and $\mathcal{O}$ a symmetric ndao over $\mathcal{L}$ s.t. $\mathcal{O}_l(\cdot,x)$ is downward closed for every $x \in L$. Then $\mathsf{WF}(\mathcal{O})$ exists, is unique, and has the following properties:*

- $\mathsf{KK}(\mathcal{O}) \preceq^A_i \mathsf{WF}(\mathcal{O})$,
- $\mathsf{WF}(\mathcal{O})$ *approximates any stable interpretation* $(x,y) \in \mathcal{S}(\mathcal{O})(x,y)$, *i.e.* $\mathsf{WF}(\mathcal{O}) \preceq^A_i (x,y)$ *for any* $(x,y) \in \mathcal{S}(\mathcal{O})(x,y)$.
- *If $\mathcal{O}$ approximates O, for any fixpoint $x \in O(x)$,* $\mathsf{WF}(\mathcal{O}) \preceq^A_i (x,x)$.

Proof. The proofs of existence and uniqueness are similar to that of Theorem 2, since $\mathcal{S}(\mathcal{O})'$ is $\preceq^A_i$-monotonic (Proposition 7).[28] The first property follows from Lemma 12, using again the results from Cousot and Cousot [19]. Notice that we need to require a symmetric ndao that is downwards closed in order to ensure $\mathsf{WF}(\mathcal{O})$ is well-defined at every step (Proposition 13).[29]

For the second property, consider some $(x,y) \in \mathcal{S}(\mathcal{O})(x,y)$. Since $(\bot,\top) \leq_i (x,y)$ and $\mathcal{S}(\mathcal{O})'$ is $\preceq^A_i$-monotonic, $\mathcal{S}(\mathcal{O})'(\bot,\top) \preceq^A_i \mathcal{S}(\mathcal{O})'(x,y)$. Let $\mathcal{S}(\mathcal{O})'(x,y) = (X,Y)$. Since (x,y) is a stable fixpoint, $x \in X$ and $y \in Y$. Thus, $\mathcal{S}(\mathcal{O})'(\bot,\top) \preceq^A_i (x,y)$. We can repeat this argument until we reach $\mathsf{KK}(\mathcal{S}(\mathcal{O})) = \mathsf{WF}(\mathcal{O})$.

The proof of the third property is similar to that of the second property. □

Example 24. We illustrate the construction in the proof of Theorem 5 by the program from Example 22. Recall that:

$$\mathcal{P} = \{p \vee q \vee r \leftarrow; \quad p \leftarrow \neg q; \quad r \leftarrow \neg p; \quad q \leftarrow \neg r\}.$$

The well-founded state is constructed as follows:

- $\mathcal{S}(\mathcal{IC}_\mathcal{P})'(\emptyset\uparrow, \{p,q,r\}\downarrow) = (\{\{p\},\{q\},\{r\}\}\uparrow, \{\{p,q,r\}\}\downarrow)$. In more detail, this is obtained as follows (recall that $C(\mathcal{IC}^l_\mathcal{P})$ is described in Example 22):

$$\begin{aligned}\mathcal{S}(\mathcal{IC}_\mathcal{P})'(\emptyset\uparrow, \{p,q,r\}\downarrow) &= \left(\bigcup_{y\in\{p,q,r\}\downarrow} C(\mathcal{IC}^l_\mathcal{P})(y)\uparrow, \bigcup_{x\in\emptyset\uparrow} C(\mathcal{IC}^l_\mathcal{P})(y)\downarrow\right)\\ &= \left(\bigcup_{y\subseteq\{p,q,r\}} C(\mathcal{IC}^l_\mathcal{P})(y)\uparrow, \bigcup_{x\subseteq\{p,q,r\}} C(\mathcal{IC}^l_\mathcal{P})(y)\downarrow\right)\end{aligned}$$

- $\left(\mathcal{S}(\mathcal{IC}_\mathcal{P})\right)'(\mathcal{S}(\mathcal{IC}_\mathcal{P})(\emptyset\uparrow, \{p,q,r\}\downarrow)) = \left(\{\{p\},\{q\},\{r\}\}\uparrow, \{\{p,q\},\{q,r\},\{p,r\}\}\downarrow\right)$.
- $\left(\mathcal{S}(\mathcal{IC}_\mathcal{P})\right)'^2(\mathcal{S}(\mathcal{IC}_\mathcal{P})(\emptyset\uparrow, \{p,q,r\}\downarrow)) = \left(\mathcal{S}(\mathcal{IC}_\mathcal{P})\right)'(\mathcal{S}(\mathcal{IC}_\mathcal{P})(\emptyset\uparrow, \{p,q,r\}\downarrow))$ and thus a fixpoint is reached.

We thus see that

$$\mathsf{WF}(\mathcal{IC}_\mathcal{P}) = \mathsf{KK}(\mathcal{S}(\mathcal{IC}_\mathcal{P})) = \left(\{\{p\},\{q\},\{r\}\}\uparrow, \{\{p,q\},\{q,r\},\{p,r\}\}\downarrow\right)$$

This is represented by the convex set:

$$\{\{p\},\{q\},\{r\},\{p,q\},\{q,r\},\{p,r\}\}$$

[28] Recall that $\mathcal{S}(\mathcal{O})'$ is obtained by taking the state-version (Remark 12) of the stable operator (Definition 20) based on $\mathcal{O}$.

[29] Investigating other conditions for well-definedness of the stable operator will result in other conditions under which this theorem can be shown.

Intuitively, the well-founded state expresses that at least one among p, q or r is true (i.e. $p \vee q \vee r$ is true), and at least one among p, q or r is false (i.e. $\neg p \vee \neg q \vee \neg r$ is true).

It is interesting to note that this is exactly the same outcome as the well-founded semantics with disjunction (see [2, Example 6]). We will see in Section 5.3 that this close resemblance is not a coincidence. For this program, the Kripke-Kleene state coincides with the well-founded state. It shows that the state semantics give meaning to programs which do not have (partial) stable interpretations. Thus, this example illustrates the existence and uniqueness-properties of the well-founded state (and the Kripke-Kleene state).

Example 25. Consider the dlp $\mathcal{P} = \{p \vee q \leftarrow \neg s; \quad s \leftarrow r; \quad r \leftarrow s\}$. We calculate $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}})$ as follows (using Theorem 4):

$$\begin{aligned}
S(\mathcal{IC}_{\mathcal{P}})'(\emptyset, \{p,q,s,r\}) &= \Big(\min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\mathcal{A}_{\mathcal{P}}})\uparrow,\ \min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\emptyset})\downarrow\Big) = \Big(\{\emptyset\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\Big)\\
S(\mathcal{IC}_{\mathcal{P}})'^2(\emptyset, \{p,q,s,r\}) &= \Big(\Big(\min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{p\}}) \cup \min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{q\}})\Big)\uparrow,\ \min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\emptyset})\downarrow\Big)\\
&= \Big(\Big(\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\Big)\\
S(\mathcal{IC}_{\mathcal{P}})'^3(\emptyset, \{p,q,s,r\}) &= \Big(\min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{p\}}) \cup \min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{q\}})\Big)\uparrow,\ \Big(\min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{p\}}) \cup \min_{\subseteq} \mathsf{Mod}(\frac{\mathcal{P}}{\{q\}})\Big)\downarrow\Big)\\
&= \Big(\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\Big)
\end{aligned}$$

and thus a fixpoint is reached after two iterations. The well-founded state is thus represented by the convex set $\{\{p\},\{q\}\}$. This can be compared to the Kripke-Kleene state $\mathsf{KK}(\mathcal{IC}_{\mathcal{P}})$:

- $(\mathcal{IC}_{\mathcal{P}})'(\emptyset, \{p,q,s,r\}) = \big(\emptyset\uparrow,\ \{\{p,s,r\},\{q,s,r\}\}\downarrow\big)$.
- $(\mathcal{IC}_{\mathcal{P}})'^2(\emptyset, \{p,q,s,r\}) = \big(\emptyset\uparrow,\ \{\{p,s,r\},\{q,s,r\}\}\downarrow\big)$ and thus a fixpoint is reached.

The Kripke-Kleene state is thus represented by the convex set $\wp(\mathcal{A}_{\mathcal{P}})$. This means that in this case, the well-founded state is significantly more precise than the Kripke-Kleene state.

This example also illustrates that the well-founded state approximates the stable interpretations of $\mathcal{P}$. Indeed, the stable interpretations are $\{p\}$ and $\{q\}$, and it holds that $(\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow) \preceq_i^A (\{p\},\{p\})$ and $(\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow) \preceq_i^A (\{q\},\{q\})$.

We conclude this section by showing that the well-founded state coincides with the well-founded fixpoint for deterministic approximation operators for approximation operators over finite lattices, thus showing that the well-founded state is a faithful generalization of the deterministic well-founded fixpoint:

Theorem 6. *Consider an ndao $\mathcal{O} : \mathcal{L}^2 \to \wp(\mathcal{L})^2$ over a finite lattice $\mathcal{L}$ s.t. $\mathcal{O}(x,y)$ is a pair of singleton sets for every $x, y \in \mathcal{L}$. Let $\mathcal{O}^{\mathsf{AFT}}$ be defined by $\mathcal{O}^{\mathsf{AFT}}(x,y) = (w,z)$ where $\mathcal{O}(x,y) = (\{w\},\{z\})$, and let $(x^{\mathsf{WF}}, y^{\mathsf{WF}})$ be the well-founded fixpoint of $\mathcal{O}^{\mathsf{AFT}}$.*[30] *Then* $\mathsf{WF}(\mathcal{O}) = (x^{\mathsf{WF}}\uparrow,\ y^{\mathsf{WF}}\downarrow)$.

Proof. Since the well-founded fixpoint of $\mathcal{O}^{\mathsf{AFT}}$ is a fixpoint of $S(\mathcal{O})$, by Proposition 11 $(x^{\mathsf{WF}}, y^{\mathsf{WF}})$ is a stable fixpoint of $S(\mathcal{O})$. By the second item of Theorem 5, $\mathsf{WF}(\mathcal{O}) \preceq_i^A (x^{\mathsf{WF}}, y^{\mathsf{WF}})$. We now show that $(x^{\mathsf{WF}}\uparrow,\ y^{\mathsf{WF}}\downarrow) \preceq_i^A \mathsf{WF}(\mathcal{O})$ by induction on the number of iterations for reaching a fixpoint. For the base case notice that, again by Proposition 11, $(\{S(\mathcal{O}_l^{\mathsf{AFT}})(\cdot,\top)\}\uparrow,\ \{S(\mathcal{O}_u^{\mathsf{AFT}})(\bot,\cdot)\}\downarrow) = S(\mathcal{O})'(\bot,\top)$. For the inductive case, suppose that for some $i \in \mathbb{N}$, (x_i, y_i) corresponds to the result of applying $S(\mathcal{O})$ i times to $(\bot,\top)$ and suppose that $(\{x_i\}\uparrow,\ \{y_i\}\downarrow) \preceq_i^A S(\mathcal{O})'^i(\bot,\top)$. This means that there are some $x', y' \in \mathcal{L}$ where x' occurs in the first component of $S(\mathcal{O})'^i(\bot,\top)$ and y' occurs in the second component of $S(\mathcal{O})'^i(\bot,\top)$ s.t. $x_i \leq x'$ and $y' \leq y_i$, i.e. $(x_i,y_i) \leq_i (x',y')$. Since $S(\mathcal{O}^{\mathsf{AFT}})$ is $\leq_i$-monotonic (see [23, Proposition 20]), $S(\mathcal{O}^{\mathsf{AFT}})(x_i,y_i) \leq_i S(\mathcal{O}^{\mathsf{AFT}})(x',y')$. By definition,

$$\begin{aligned}
(S(\mathcal{O})')^{i+1}(\bot,\top) &= \Big(\bigcup_{(x,y)\in S(\mathcal{O})'^i(\bot,\top)} C(\mathcal{O}_l)(y)\uparrow,\ \bigcup_{(x,y)\in S(\mathcal{O})'^i(\bot,\top)} C(\mathcal{O}_u)(x)\downarrow\Big)\\
&= \Big(\bigcup_{(x,y)\in S(\mathcal{O})'^i(\bot,\top)} \{C(\mathcal{O}_l^{\mathsf{AFT}})(y)\}\uparrow,\ \bigcup_{(x,y)\in S(\mathcal{O})'^i(\bot,\top)} \{C(\mathcal{O}_u^{\mathsf{AFT}})(x)\}\downarrow\Big)
\end{aligned}$$

and thus $(S(\mathcal{O}^{\mathsf{AFT}}))^{i+1}(x',y') \preceq_i^A S(\mathcal{O})'^{i+1}(\bot,\top)$. Hence, with a slight abuse of the notations, we have shown that $S(\mathcal{O}^{\mathsf{AFT}})(x_i,y_i) \preceq_i^A S(\mathcal{O})'^{i+1}(\bot,\top)$. □

An interesting property of the well-founded state of $\mathcal{IC}_{\mathcal{P}}$ is that for positive logic programs, the well-founded state coincides with the *minimal* models of the logic program.

[30] Recall Definition 6 for the definition of the well-founded fixpoint of a deterministic approximation operator.

Proposition 15. *If $\mathcal{P}$ is a positive dlp, then* $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}}) = (\min_{\subseteq} mod(\mathcal{P})\uparrow, \min_{\subseteq} mod(\mathcal{P})\downarrow)$.

Proof. Notice first that, since $\mathcal{P}$ is positive, $\frac{\mathcal{P}}{y} = \mathcal{P}$ for any $y \subseteq \mathcal{A}_{\mathcal{P}}$ (no rule in $\mathcal{P}$ contains negative literals in the body, and so no rule is deleted or transformed in the construction of $\frac{\mathcal{P}}{y}$). By Proposition 10, this means that, for any $y \subseteq \mathcal{A}_{\mathcal{P}}$, $C(\mathcal{IC}^l_{\mathcal{P}})(y) = \min_{\subseteq}(mod(\mathcal{P}))$. Thus, $S(\mathcal{IC}_{\mathcal{P}})'(\emptyset, \mathcal{A}_{\mathcal{P}}) = (\min_{\subseteq} mod(\mathcal{P})\uparrow, \min_{\subseteq} mod(\mathcal{P})\downarrow)$ and a fixpoint is reached at the first iteration of the construction of the well-founded state. □

5.3. The relationship with well-founded semantics with disjunction

As another indication for the usefulness of our constructions, we show that the well-founded state approximates the existing well-founded semantics for disjunctive logic programs, which is the well-founded semantics for disjunction WFS_d by Alcântara, Damásio and Pereira [2].

Alcântara, Damásio and Pereira [2] define the well-founded semantics for disjunction WFS_d which bears similarities to our well-founded semantics. We present this semantics here, adapting the notation and technicalities to our setting.

The basic idea behind the well-founded semantics with disjunction as defined by Alcântara, Damásio and Pereira [2] is the following: a sequence of sets of lower bounds and upper bounds approximating the stable models of a disjunctive logic program is constructed iteratively. Given a set of lower bounds and a set of upper bounds, new (more precise) lower and upper bounds can be obtained by applying the two-valued immediate consequence operator IC to the reducts of the program obtained on the basis of the upper bounds and lower bounds (respectively). The lower bounds are thus used to obtain new upper bounds and vice versa. This iteration leads to more and more precise lower and upper bounds, until a fixpoint is reached. This well-founded semantics is guaranteed to exist, be unique, and coincide with the well-founded model for normal logic programs [2].

We now develop this notion in more technical details. The operator $\Gamma_{\mathcal{P}}(X)$ is defined as follows[31]:

$$\Gamma_{\mathcal{P}}(X) = \bigcup_{x \in X} \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{x}))$$

Instead of just applying the $\Gamma_{\mathcal{P}}$-operator to the lower bound and closing it downwards to obtain a new upper bound, a form of closed world reasoning is performed by Alcântara, Damásio and Pereira [2] by applying $\Gamma_{\mathcal{P}}$ to the interpretations in the lower bound not containing any atoms not occurring in any upper bound. Formally, this is done by defining the set $\mathfrak{F}(Y)$ that contains all atoms not occurring in any upper bound $y \in Y$, and the set $\mathfrak{T}_Y(X)$ that contains all the sets in X without any atoms in $\mathfrak{F}(Y)$.

Definition 23. Given $X, Y \subseteq \mathcal{A}_{\mathcal{P}}$, we define:

- $\mathfrak{F}(Y) = \{\alpha \in \mathcal{A}_{\mathcal{P}} \mid \alpha \notin \bigcup Y\}$.
- $\mathfrak{T}_Y(X) = \{x \in X \mid x \cap \mathfrak{F}(Y) = \emptyset\}$.

We are now ready to define the Φ-operator which is the basis of the well-founded semantics for disjunction [2], obtained by taking as set of new lower bounds all the models of reducts of some upper bound, and as new upper bounds all the models of reducts of some lower bound in $\mathfrak{T}_Y(X)$.

Definition 24. Let some disjunctive logic program $\mathcal{P}$ and some $X, Y \subseteq \wp(\mathcal{A}_{\mathcal{P}})$ be given. Then:

$$\Phi_{\mathcal{P}}(X, Y) = (\Gamma_{\mathcal{P}}(Y)\uparrow, \Gamma_{\mathcal{P}}(\mathfrak{T}_Y(X))\downarrow).$$

The *well-founded semantics for disjunction of a program* $\mathcal{P}$, denoted $\mathsf{WFS}_d(\mathcal{P})$ is defined as the least fixpoint of $\Phi_{\mathcal{P}}$, obtained by applying $\Phi_{\mathcal{P}}$ iteratively to $(\emptyset, \mathcal{A}_{\mathcal{P}})$.

We can show that our well-founded semantics is an approximation of the WFS_d. It really is an approximation, since we do not apply closed world reasoning in constructing the lower bound (but one could, as we demonstrate below).

Theorem 7. *For any disjunctive logic program $\mathcal{P}$,* $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}}) \preceq^A_i \mathsf{WFS}_d(\mathcal{P})$.

The following example (taken from [2]) shows that $\mathsf{WFS}_d(\mathcal{P})$ can give rise to a strictly more precise approximation than that of $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}})$:

Example 26. Consider the logic program $\mathcal{P} = \{p \vee q \leftarrow;\ \ q \leftarrow \neg r\}$. We first calculate $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}})$ as follows:

$$S(\mathcal{IC}_{\mathcal{P}})(\{\emptyset\}\uparrow, \{p,q,r\}\downarrow) = \left(\min_{\subseteq}\Big(\bigcup_{x \subseteq \{p,q,r\}} mod_2(\frac{\mathcal{P}}{x})\Big)\uparrow,\ \min_{\subseteq}\Big(\bigcup_{x \subseteq \{p,q,r\}} mod_2(\frac{\mathcal{P}}{x})\Big)\downarrow\right)$$

[31] Recall that the *two-valued models* of a positive program $\mathcal{P}$ are the sets $x \subseteq \mathcal{A}_{\mathcal{P}}$ s.t. for every $\bigvee \Delta \leftarrow \phi \in \mathcal{P}$, $(x,x)(\phi) = \mathsf{T}$ implies $x \cap \Delta \neq \emptyset$ (Footnote 6).

$$= \big(\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\big)$$

$$S^2(\mathcal{IC}_{\mathcal{P}})(\{\emptyset\}\uparrow,\{p,q,r\}\downarrow) = \left(\min_{\subseteq}\big(mod_2(\frac{\mathcal{P}}{\{p\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}})\big)\uparrow,\ \min_{\subseteq}\big(\bigcup_{x\subseteq\{p,q,r\}} mod_2(\frac{\mathcal{P}}{x})\big)\downarrow\right)$$

$$= \big(\{\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\big)$$

$$S^3(\mathcal{IC}_{\mathcal{P}})(\{\emptyset\}\uparrow,\{p,q,r\}\downarrow) = \left(\min_{\subseteq}\big(mod_2(\frac{\mathcal{P}}{\{q\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}})\big)\uparrow,\ \min_{\subseteq}\big(\bigcup_{x\subseteq\{p,q,r\}} mod_2(\frac{\mathcal{P}}{x})\big)\downarrow\right)$$

$$= \big(\{\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow\big)$$

and so a fixpoint is reached after the second iteration. The well-founded state for this program thus corresponds to the convex set $\{\{q\},\{p\},\{q\}\}$.

It can be observed that $\frac{\mathcal{P}}{x} = \{p \vee q \leftarrow\}$ for any $\{r\} \subseteq x \subseteq \mathcal{A}_{\mathcal{P}}$. In particular, as there are some $x \in \{p\},\{q\}\uparrow$ s.t. $r \in x$, this explains why $\{p\}$ is part of the upper bound of $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}})$. It is exactly this kind of behaviour that the filtering out lower bounds that have elements not occurring in any upper bound in $\mathfrak{T}_Y(X)$ tries to avoid. Indeed, $\mathsf{WFS}_d(\mathcal{P})$ is built up as follows:

$$\Phi_{\mathcal{P}}(\{\emptyset\},\{\{p,q,r\}\}) = S(\mathcal{IC}_{\mathcal{P}})(\{\emptyset\}\uparrow,\{p,q,r\}\downarrow)$$

$$= (\{\{p\},\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow)$$

$$\Phi^2_{\mathcal{P}}(\{\emptyset\},\{\{p,q,r\}\}) = \left(\min_{\subseteq}\big(mod_2(\frac{\mathcal{P}}{\{p\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}})\big)\uparrow,\ \min_{\subseteq}\big(mod_2(\frac{\mathcal{P}}{\{p\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}}) \cup mod_2(\frac{\mathcal{P}}{\{q,r\}})\big)\downarrow\right)$$

$$= \big(\{\{p\}\}\uparrow,\ \{\{p\}\}\downarrow\big)$$

$$\Phi^3_{\mathcal{P}}(\{\emptyset\},\{\{p,q,r\}\}) = \left(\min_{\subseteq}\Big(mod_2(\frac{\mathcal{P}}{\{p\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}})\Big)\uparrow,\ \min_{\subseteq}\Big(mod_2(\frac{\mathcal{P}}{\{p\}}) \cup mod_2(\frac{\mathcal{P}}{\{q\}}) \cup mod_2(\frac{\mathcal{P}}{\{q,r\}})\Big)\downarrow\right)$$

$$= \big(\{\{p\}\}\uparrow,\ \{\{p\}\}\downarrow\big)$$

The upper bound of $\Phi^2_{\mathcal{P}}(\{\emptyset\},\{\{p,q,r\}\})$ can be seen to hold in view of $\mathfrak{T}_{\{\{p\},\{q\}\}\Downarrow}(\{\{p\},\{q\}\}\uparrow) = \{\{p\},\{q\},\{p,q\}\}$. After two iterations, a fixpoint is reached, and thus we see that $\mathsf{WFS}_d(\mathcal{P})$ corresponds to the convex set $\{\{p\}\}$, which is a more precise approximation than $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}})$. Formally:

$$\mathsf{WF}(\mathcal{IC}_{\mathcal{P}}) = (\{\{q\}\}\uparrow,\ \{\{p\},\{q\}\}\downarrow) <_i^A \mathsf{WFS}_d(\mathcal{P}) = (\{\{p\}\}\uparrow,\ \{\{p\}\}\downarrow).$$

Indeed, it is not so hard to generalize the $\mathfrak{T}_Y(X)$-construction algebraically. We start by observing that the formulation by Alcântara, Damásio and Pereira [2] can be significantly simplified. Firstly, we observe that $\mathfrak{F}(Y) = \mathcal{A}_{\mathcal{P}} \setminus \bigcup Y$. $\mathfrak{T}_Y(X)$ can then be simplified as $\mathfrak{T}_Y(X) = \{x \in X \mid x \subseteq \bigcup Y\}$. This is generalized algebraically to $\mathfrak{T}_Y(X) = \{x \in X \mid x \subseteq \bigsqcup Y\}$. Notice that this requires the assumption that the lattice in question is complete, as $\bigsqcup Y$ is not guaranteed to exist otherwise.

On the basis of this, we can define $\mathcal{O}'_{\mathsf{cw}}(X,Y) = \mathcal{O}'_l(\mathfrak{T}_Y(X),Y)$, and show the following result:

Proposition 16. *Let $\mathcal{O}'$ be an ndso approximating an operator O over a complete lattice $\langle\mathcal{L},\leq\rangle$. Then $\mathcal{O}'_{\mathsf{cw}}$ is an ndso approximating O such that for any $\mathbf{X} \in \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$, $\mathcal{O}'(\mathbf{X}) \preceq_i^A \mathcal{O}'_{\mathsf{cw}}(\mathbf{X})$.*

Proof. We first note that †: $(X,Y) \preceq_i^A (\mathfrak{T}_Y(X),Y)$. This is immediate from the fact that $\mathfrak{T}_Y(X) \subseteq X$.

$\preceq_i^A$-monotonicity follows from †.

We now show that $\mathcal{O}'_{\mathsf{cw}}$ approximates O. Indeed, as $\mathfrak{T}_{\{z\}}(\{z\}) = \{z\}$, and $\mathcal{O}'$ approximates O, it holds that $\mathcal{O}'_{\mathsf{cw}}(\{z\},\{z\}) = \mathcal{O}'(\{z\},\{z\}) = (O(z)\uparrow, O(z)\downarrow)$.

We now show that for any $(X,Y) \in \wp_\uparrow(\mathcal{L}) \times \wp_\downarrow(\mathcal{L})$, $\mathcal{O}'(X,Y) \preceq_i^A \mathcal{O}'_{\mathsf{cw}}(X,Y)$. This is immediate from the fact that $\mathcal{O}'$ is $\preceq_i^A$-monotonic and from †, as $(X,Y) \preceq_i^A (\mathfrak{T}_Y(X),Y)$ implies $\mathcal{O}'(X,Y) \preceq_i^A \mathcal{O}'(\mathfrak{T}_Y(X),Y) = \mathcal{O}'_{\mathsf{cw}}(X,Y)$. □

The well-founded state taking into account the closed world assumption can now be defined as the $\preceq_i^A$-least fixpoint of $(S(\mathcal{O}))'_{\mathsf{cw}}$, and is denoted by $\mathsf{WFS}_{\mathsf{cw}}(\mathcal{O})$. Applying this construction to $\mathcal{IC}_{\mathcal{P}}$ allows to represent the well-founded semantics for disjunction by Alcântara, Damásio and Pereira [2]:

Proposition 17. *For any disjunctive logic program $\mathcal{P}$, $\mathsf{WF}_{\mathsf{cw}}(\mathcal{IC}_{\mathcal{P}}) = \mathsf{WFS}_d(\mathcal{P})$.*

Proof. This follows immediately from the fact that $(S(\mathcal{O}))'_{\mathsf{cw}} = \Phi_{\mathcal{P}}$, which is straightforward from the definition of the closed world assumption (Indeed, the construction is an algebraic generalization of that by Alcântara, Damásio and Pereira [2]). □

An interesting line of research would be to conduct a more systematic investigation of ways of adapting or even constructing ndsos. This, however, is outside the scope of this paper.

Summary We summarize the results in Section 5. We first defined and studied stable operators (Section 5.1), establishing which properties carry over (under certain conditions) from the deterministic to the non-deterministic setting, and showing the usefulness of stable fixpoints in disjunctive logic programming. Then (Section 5.2), we introduced and studied the well-founded state, proving its existence, uniqueness, and showing that it is more precise than the Kripke-Kleene state and that it approximates any fixpoint of the approximated operator. Finally (Section 5.3), we have shown that the well-founded state is useful for knowledge representation, as it is closely related to the well-founded semantics for disjunction [2].

6. Related work

The starting point of this work is the approximation fixpoint theory (for deterministic operator), as introduced by Denecker, Marek and Truszczyński [23], followed by a series of papers [3,9,12,18,21,22,24,37,54]. As indicated previously (see, e.g. Remark 5), this work generalizes AFT in the sense that all the operators and fixpoints defined in this paper coincide with the respective counterparts for deterministic operators.

This paper (extending and improving our paper [38]) is also inspired by the work of Pelov and Truszczyński [48], which extends approximation fixpoint theory to dealing with non-deterministic operators. Their work provides a representation theorem, in terms of non-deterministic AFT, of specific two-valued semantics for disjunctive logic programs (namely, the two-valued stable semantics and two-valued weakly supported and supported models). We have compared our work to that of Pelov and Truszczyński [48] in Section 1.

To the best of our knowledge, the only setting with a similar unifying potential that has been applied to non-deterministic or disjunctive reasoning is *equilibrium logic* [46]. The similarities between equilibrium logic and AFT have been noted before [21], where it was indicated that equilibrium semantics are defined for a larger class of logic programs than those that are represented by AFT, a limitation of AFT which we have overcome in this paper. Furthermore, defining three-valued stable and well-founded semantics is not possible in standard equilibrium logic, but requires an extension known as *partial equilibrium logic* [16,17], which can be seen as a six-valued semantics. In contrast, the well-founded semantics *is defined* in AFT using the same operator used to define the stable semantics. That being said, in future work we plan to compare the well-founded semantics for DLP obtained on the basis of partial equilibrium logic and the well-founded semantics obtained in this work in more detail.

7. Summary, conclusion, and future work

This paper contains a full generalization of approximation fixpoint theory to non-deterministic operators. We introduced deterministic operators, their non-deterministic approximation operators, and the various fixpoint semantics, namely, the Kripke-Kleene interpretation and state semantics, the stable interpretation semantics and the well-founded state semantics. The properties of these semantics and their representation of disjunctive logic programming semantics are summarized in Table 4. The relation between these fixpoint semantics is summarized in Fig. 3.

Table 4
Approximation operators and their properties.

Name	Definition	Exist	Unique	$\leq_t$-minimality	Result	DLP-representation
KK interpetation	$\mathrm{lfp}(\mathcal{O})$	×	×	×		weakly supported (Theorem 1)
KK state	$\mathrm{lfp}(\mathcal{O}')$	✓	✓	×	Theorem 2	
Stable interpetation	$\mathrm{fp}(S(\mathcal{O}))$	×	×	✓	Proposition 14	stable models (Theorem 4)
Well-founded state	$\mathrm{lfp}(S(\mathcal{O})')$	✓	✓	×	Theorem 5	WF sem. with disjunct. (Theorem 7) Min. mod. of positive dlps (Proposition 15)

This work also allows to generalize the results in [3,48], which provide further approximation operators for disjunctive logic programs with aggregates or external atoms, to additional semantics of disjunctive logic programs, thus answering an open question in these works.

The advantage of studying non-deterministic operators is thus at least twofold:

1. allowing to define a family of semantics for non-monotonic reasoning with disjunctive information,
2. clarifying similarities and differences between semantics stemming from the use of different operators.

The introduction of disjunctive information in AFT points to a wealth of further research, such as defining three-valued and well-founded semantics for various disjunctive nonmonotonic formalisms and studying on the basis of which operators various well-founded semantics for DLP can be represented in our framework. For example, our framework has been used for characterising existing semantics and defining new semantics for disjunctive logic programs with aggregates in the body [39] and logic programs with choice constructs in the head [36]. This framework can also potentially be used for defining three-valued and well-founded semantics for propositional theories [57], logic programs with forks [1], and disjunctive default logics [14,33].

Non-deterministic approximation fixpoint theory has already been applied in order to obtain a generalization of abstract dialectical frameworks (ADFs) to conditional abstract dialectical frameworks. In a nutshell, abstract dialectical frameworks consist of sets of

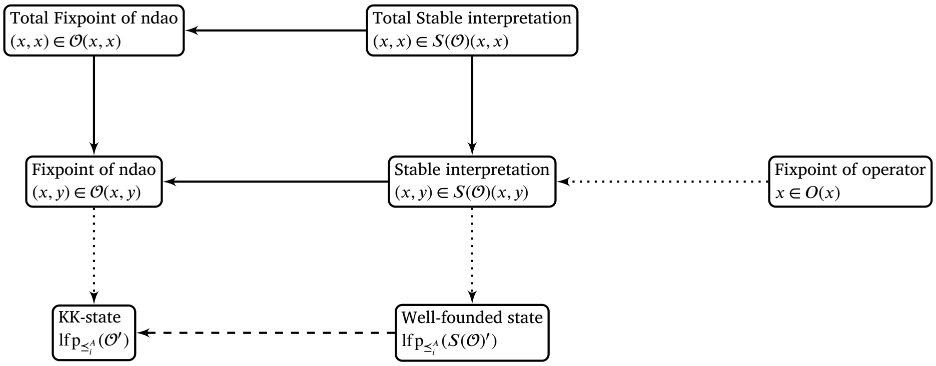


Fig. 3. Relations between various fixpoints introduced in this paper. The arrows have the following meaning: full arrows mean that every instance of the first block (i.e., at the outgoing end of the arrow) is an instance of the second block (at the ingoing end of the arrow). Dotted arrows mean that all instances of the first block are approximated by the second block. Dashed arrows mean that every instance of the first block is more precise than every instance of the second block. The relations are shown in Theorem 3 and Theorem 5.

arguments or atoms, in which every atom p is assigned a Boolean acceptance condition C_p, which codifies under which conditions an atom can be accepted. Heyninck and co-authors [40] generalized ADFs to allow for acceptance conditions to be assigned to any, i.e. possibly non-atomic, formulas. This necessitated the generalization of the so-called Γ-operator to a non-deterministic operator, which was done in the framework of non-deterministic approximation fixpoint theory as presented here.

Our framework lays the ground for the generalization to a non-deterministic setting of various interesting concepts introduced (or adapted) to AFT, such as ultimate approximations [22] (already generalized to non-deterministic AFT by Heyninck and Bogaerts [39]), grounded fixpoints [12], strong equivalence [56], stratification [60] and argumentative representations [37]. Extensions to DLP with negations in the rules' heads and corresponding 4-valued semantics [50] can also be considered.

Another issue that is worth some consideration is a study of the complexity of computing different types of fixpoints of non-deterministic approximation operators, in the style of Strass and Wallner [54].

CRediT authorship contribution statement

Jesse Heyninck: Conceptualization, Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Ofer Arieli:** Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Bart Bogaerts:** Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jesse Heyninck reports financial support was provided by German National Science Foundation (DFG-project KE-1413/11-1). Ofer Arieli reports financial support was provided by Israel Science Foundation (Grant 550/19). Jesse Heyninck reports financial support was provided by Fonds Wetenschappelijk Onderzoek Vlaanderen (Grant G0B2221N). Bart Bogaerts reports was provided by Flemish Government (FLAIR).

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank the reviewers of the conference version of this paper [38], as well as of this paper, for their diligent reviews and helpful remarks. The first author was partially supported by the German National Science Foundation (DFG-project KE-1413/11-1). He worked on this paper while affiliated with the Vrije Universiteit Brussel, the Technische Universität Dortmund, and CAIR South-Africa, in addition to his current affiliation. The second author is supported by the Israel Science Foundation (Grant 550/19). The first and last author are supported by the Flemish Government in the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme (FLAIR), and by the FWO Flemish Government project G0B2221N.

Appendix A. Proofs of results on disjunctive logic programming

Proposition 2. $\mathcal{IC}_{\mathcal{P}}$ is a symmetric ndao that approximates $IC_{\mathcal{P}}$.

Proof. It is clear that for any $x \in \mathcal{L}$, $\mathcal{HD}^l_{\mathcal{P}}(x,x) = \mathcal{HD}^u_{\mathcal{P}}(x,x) = HD_{\mathcal{P}}(x,x)$ (as for any ϕ, $(x,x)(\phi) \in \{\mathsf{T},\mathsf{F}\}$). Thus, $\mathcal{IC}_{\mathcal{P}}$ approximates $IC_{\mathcal{P}}$ and is exact. We now show that it is $\preceq^A_i$-monotonic. For this, consider some $(x_1,y_1) \leq_i (x_2,y_2)$. We show by induction on ϕ that if $(x_1,y_1)(\phi) \geq_t \mathsf{C}$ then $(x_2,y_2)(\phi) \geq_t \mathsf{C}$. The base case is clear as $\phi \in x_1$ and $x_1 \subseteq x_2$ implies $\phi \in x_2$. For the inductive case, notice that the cases for $\phi = \phi_1 \wedge \phi_2$ and $\phi = \phi_1 \vee \phi_2$ follow immediately from the inductive hypothesis. Suppose now that $\phi = \neg\phi_1$. $(x_1,y_1)(\neg\phi_1) \geq_t \mathsf{C}$ means that $\phi_1 \notin y_1$. Since $y_2 \subseteq y_1$, also $\phi_1 \notin y_2$.

We now show $\preceq^A_i$-monotonicity, which follows immediately from $\mathcal{HD}^l_{\mathcal{P}}(x_1,y_1) \subseteq \mathcal{HD}^l_{\mathcal{P}}(x_2,y_2)$. To see the latter, suppose that $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x_1,y_1)$, i.e. for some $\bigvee\Delta \leftarrow \phi$, $(x_1,y_1)(\phi) \geq_t \mathsf{C}$. Then $(x_2,y_2)(\phi) \geq_t \mathsf{C}$ and thus $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x_2,y_2)$.

We finally show that $\mathcal{IC}_{\mathcal{P}}$ is symmetric. For this, we show the following lemma:

Lemma 13. *For any $x, y \subseteq \mathcal{A}_{\mathcal{P}}$:*

1. $(x,y)(\phi) = \mathsf{T}$ *iff* $(y,x)(\phi) = \mathsf{T}$*,*
2. $(x,y)(\phi) = \mathsf{F}$ *iff* $(y,x)(\phi) = \mathsf{F}$*,*
3. $(x,y)(\phi) = \mathsf{C}$ *iff* $(y,x)(\phi) = \mathsf{U}$*.*

Proof. We show the third item, the first two items are shown similarly. We show this by induction on the structure of ϕ. For the base case, let $\phi \in \mathcal{A}_{\mathcal{P}}$. Then $(x,y)(\phi) = \mathsf{C}$ iff $\phi \in x \setminus y$, and, since $(y,x)(\phi) = \mathsf{U}$ iff $\phi \in x \setminus y$, the base case is proven. For the inductive case, notice that the cases where $\phi = \phi_1 \vee \phi_2$ or $\phi = \phi_1 \wedge \phi_2$ follow immediately from the inductive hypothesis. Suppose that $\phi = \neg\phi_1$. Since $(x,y)(\neg\phi_1) = \mathsf{C}$ iff $(x,y)(\phi_1) = \mathsf{C}$, and $(y,x)(\neg\phi_1) = \mathsf{U}$ iff $(y,x)(\phi_1) = \mathsf{U}$, we obtain by the induction hypothesis that $(x,y)(\neg\phi_1) = \mathsf{C}$ iff $(y,x)(\neg\phi_1) = \mathsf{U}$. ■

From Lemma 13 it follows that $\mathcal{HD}^l_{\mathcal{P}}(x,y) = \mathcal{HD}^u_{\mathcal{P}}(y,x)$ and thus $\mathcal{IC}^l_{\mathcal{P}}(x,y) = \mathcal{IC}^u_{\mathcal{P}}(y,x)$. □

Proposition 2. $\mathcal{IC}_{\mathcal{P}}$ is a symmetric ndao that approximates $IC_{\mathcal{P}}$.

Proof. It is clear that $\mathcal{IC}^{\mathsf{DMT}}_{\mathcal{P}}$ approximates $IC_{\mathcal{P}}$, as $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x,x) = \mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x,x) = IC^{\mathsf{DMT}}_{\mathcal{P}}(x,x)$ for any $x \subseteq \mathcal{A}_{\mathcal{P}}$. We now show it is $\preceq^A_i$-monotonic. Consider some $x_1 \subseteq x_2 \subseteq y_2 \subseteq y_1$. We show that $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_1,y_1) \subseteq \mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_2,y_2)$ and $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x_2,y_2) \subseteq \mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x_1,y_1)$, which immediately implies that $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_1,y_1) \preceq^S_L \mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_2,y_2)$ and $\mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x_2,y_2) \preceq^H_L \mathcal{HD}^{\mathsf{DMT},u}_{\mathcal{P}}(x_1,y_1)$. To see that $\mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_1,y_1) \subseteq \mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_2,y_2)$, consider some $\Delta \in \mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_1,y_1)$. Then for every $z \in [x_1,y_2]$, there is some $\bigvee\Delta \leftarrow \phi$ s.t. $(z,z)(\phi) = \mathsf{T}$. Since $[x_2,y_2] \subseteq [x_1,y_1]$, also $\Delta \in \mathcal{HD}^{\mathsf{DMT},l}_{\mathcal{P}}(x_2,y_2)$. The other claim is analogous. □

Theorem 1. Given a dlp $\mathcal{P}$ and a consistent interpretation $(x,y) \in (\wp(\mathcal{A}_{\mathcal{P}}))^2$, it holds that (x,y) is a weakly supported model of $\mathcal{P}$ iff $(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$.

Proof. [⇒] Suppose that (x,y) is weakly supported. We first show that for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$, $\Delta \cap x \neq \emptyset$. Indeed, let $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$, i.e. $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ and $(x,y)(\phi) = \mathsf{T}$ (notice that since (x,y) is consistent, $(x,y)(\phi) \neq \mathsf{C}$). Since (x,y) is a model of $\mathcal{P}$, $(x,y)(\bigvee\Delta) \geq_t \mathsf{T}$, and so $\Delta \cap x \neq \emptyset$.

We now show that $y \cap \mathcal{HD}^u_{\mathcal{P}}(x,y) \neq \emptyset$. Suppose for this that $(y,x)(\phi) \in \{\mathsf{C},\mathsf{T}\}$ for some $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$. We first show the following lemma:

Lemma 14. *For any formula ϕ and any $x, y \subseteq \mathcal{A}_{\mathcal{P}}$, $(y,x)(\phi) \in \{\mathsf{T},\mathsf{C}\}$ implies that $(x,y)(\phi) \in \{\mathsf{T},\mathsf{U}\}$ and $(y,x)(\phi) \in \{\mathsf{F},\mathsf{C}\}$ implies that $(x,y)(\phi) \in \{\mathsf{F},\mathsf{U}\}$.*

Proof. We show this by induction on the structure of ϕ. For the base case, suppose that $\phi \in \mathcal{A}_{\mathcal{P}}$. If $(y,x)(\phi) \in \{\mathsf{T},\mathsf{C}\}$, then $\phi \in y$ and thus $(x,y)(\phi) \in \{\mathsf{T},\mathsf{U}\}$. Likewise, $(y,x)(\phi) \in \{\mathsf{F},\mathsf{C}\}$ means that $\phi \notin x$ and thus $(x,y)(\phi) \in \{\mathsf{F},\mathsf{U}\}$. For the inductive case, suppose that the claim holds for ϕ and ψ. If $(y,x)(\neg\phi) \in \{\mathsf{T},\mathsf{C}\}$, then $(y,x)(\phi) \in \{\mathsf{F},\mathsf{C}\}$, and with the inductive hypothesis, $(x,y)(\phi) \in \{\mathsf{F},\mathsf{U}\}$, which implies that $(y,x)(\neg\phi) \in \{\mathsf{T},\mathsf{U}\}$. The other cases are similar. ■

By Lemma 14 and since $\Delta \in \mathcal{HD}^u_{\mathcal{P}}(x,y)$ implies there is some $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ with $(y,x)(\phi) \in \{\mathsf{T},\mathsf{C}\}$, it follows that for every $\Delta \in \mathcal{HD}^u_{\mathcal{P}}(x,y)$, $\Delta \cap y \neq \emptyset$.

It remains to be shown that $x \subseteq \bigcup \mathcal{HD}^l_{\mathcal{P}}(x,y)$, and $y \subseteq \bigcup \mathcal{HD}^l_{\mathcal{P}}(y,x) = \bigcup \mathcal{HD}^u_{\mathcal{P}}(x,y)$, which is immediate from the fact that since (x,y) is weakly supported, for every atom that is true (respectively undecided) we can find a rule whose body is true (respectively undecided) that has this atom in the head.

[⇐] Suppose that $(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$. We first show that (x,y) is a model of $\mathcal{P}$. Indeed, suppose that for $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$, $(x,y)(\phi) = \mathsf{T}$. Then $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$ and thus (since $(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$), $\Delta \cap x \neq \emptyset$, i.e., $(x,y)(\bigvee\Delta) = \mathsf{T}$. The case for $(x,y)(\phi) = \mathsf{U}$ is similar and the case for $(x,y)(\phi) = \mathsf{F}$ is trivial. We now show that (x,y) is weakly supported. As $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$, we know that $p \in \bigcup \mathcal{HD}^l_{\mathcal{P}}(x,y)$, which implies that there is a $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ s.t. $(x,y)(\phi) = \mathsf{T}$. The proof for $p \in y$ is similar. □

Proposition 10. Consider a dlp $\mathcal{P}$ and some $y \subseteq \mathcal{A}_{\mathcal{P}}$. Then $C(\mathcal{IC}_{\mathcal{P}})(y) = \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{y}))$.

Proof. We first show that $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$ implies $x \in mod_2(\frac{\mathcal{P}}{y})$. Indeed, suppose that $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$ and consider some $\bigvee\Delta \leftarrow \bigwedge_{i=1}^n \alpha_i \wedge \bigwedge_{j=1}^m \neg\beta_j$. If $\beta_j \in y$ for some $j = 1,\dots,m$, then $(x,y)(\neg\beta_j) \in \{\mathsf{U},\mathsf{F}\}$ and thus we can ignore this rule. Suppose then that $\beta_j \notin y$ for $j = 1,\dots,m$. Suppose now that $(x,y)(\alpha_i) \in \{\mathsf{T},\mathsf{C}\}$ for $i = 1,\dots,n$, i.e., $\alpha_i \in x$ for $i = 1,\dots,n$. Then $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$ and thus, since $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$, $x \cap \Delta \neq \emptyset$.

We now show that $x \in mod_2(\frac{\mathcal{P}}{y})$ implies that $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$. Indeed, consider some rule of the form $\bigvee\Delta \leftarrow \bigwedge_{i=1}^n \alpha_i \in \frac{\mathcal{P}}{y}$, i.e. there is some $\bigvee\Delta \leftarrow \bigwedge_{i=1}^n \alpha_i \wedge \bigwedge_{j=1}^m \neg\beta_j \in \mathcal{P}$ s.t. $\beta_j \notin y$ for $j = 1,\dots,m$. If $\alpha_i \notin x$ for some $i = 1,\dots,n$ then we can safely ignore the rule. Suppose thus that $\alpha_i \in x$ for some $i = 1,\dots,n$. Then $(x,y)(\alpha_i) \in \{\mathsf{T},\mathsf{C}\}$ for $i = 1,\dots,n$ and $(x,y)(\beta_j) \in \{\mathsf{T},\mathsf{C}\}$ for $j = 1,\dots,m$ and therefore, as $x \in mod_2(\frac{\mathcal{P}}{y})$, $x \cap \Delta \neq \emptyset$. This means we have shown that for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$, $\Delta \cap x \neq \emptyset$. Thus, $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$.

We now show that $x \in \mathrm{lfp}(\mathcal{IC}^l_{\mathcal{P}}(\cdot,y))$ implies $x \in \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{y}))$. Indeed, suppose $x \in \mathrm{lfp}(\mathcal{IC}^l_{\mathcal{P}}(\cdot,y))$. Since this means that $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$, with the first item, $x \in mod_2(\frac{\mathcal{P}}{y})$. Suppose towards a contradiction there is some $x' \subset x$ s.t. $x' \in mod_2(\frac{\mathcal{P}}{y})$. Then with the second item, $x' \in \mathcal{IC}^l_{\mathcal{P}}(x',y)$, contradicting $x \in \mathrm{lfp}(\mathcal{IC}^l_{\mathcal{P}}(\cdot,y))$.

The proof that $x \in \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{y}))$ implies $x \in \mathrm{lfp}(\mathcal{IC}^l_{\mathcal{P}}(\cdot,y))$ is analogous. □

Proposition 12. For any dlp $\mathcal{P}$ and any $y \subseteq \mathcal{A}_{\mathcal{P}}$, $\mathcal{IC}_{\mathcal{P}}(\cdot,y)$ is downward closed.

Proof. Let $\{x_\epsilon\}_{\epsilon<\alpha}$ be a descending chain of sets of atoms of post-fixpoints of $\mathcal{IC}^l_{\mathcal{P}}(\cdot,y)$ for some $y \subseteq \mathcal{A}_{\mathcal{P}}$, and let $x = \bigcap\{x_\epsilon\}_{\epsilon<\alpha}$. We show that $\mathcal{IC}_{\mathcal{P}}(x,y) \preceq^S_L \{x\}$. If the chain is finite, this is trivial. Suppose therefore that $\{x_\epsilon\}_{\epsilon<\alpha}$ is infinite.

We first show the following lemma:

Lemma 15. $(x,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$ *implies* $(x_\epsilon,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$ *for every* $\epsilon \prec \alpha$, *and* $(x,y)(\phi) \in \{\mathsf{F},\mathsf{C}\}$ *implies* $(x_\epsilon,y)(\phi) \in \{\mathsf{F},\mathsf{C}\}$ *for every* $\epsilon \prec \alpha$.

Proof. By induction on the structure of ϕ. Base case: suppose that $\phi = p \in \mathcal{A}_{\mathcal{P}}$. If $(x,y)(p) \in \{\mathsf{T},\mathsf{C}\}$ then $p \in x$ and so $p \in x_\epsilon$ for every $\epsilon \prec \alpha$, thus $(x_\epsilon,y)(p) \in \{\mathsf{T},\mathsf{C}\}$ as well. If $(x,y)(p) \in \{\mathsf{F},\mathsf{C}\}$ then $p \notin y$, and so $(x_\epsilon,y)(p) \in \{\mathsf{F},\mathsf{C}\}$ as well. Inductive case: the cases where $\phi = p_1 \wedge p_2$ and $\phi = p_1 \vee p_2$ are straightforward. Suppose now that $\phi = \neg p$ and $(x,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$. Thus, $(x,y)(p) \in \{\mathsf{F},\mathsf{C}\}$ and by the inductive hypothesis, $(x_\epsilon,y)(p) \in \{\mathsf{F},\mathsf{C}\}$ for every $\epsilon < \alpha$, which implies that $(x_\epsilon,y)(\neg p) \in \{\mathsf{T},\mathsf{C}\}$ for every $\epsilon \prec \alpha$. The proof of the other case is similar. ■

Back to the proof of the proposition. We first show that $x \cap \Delta \neq \emptyset$ for every $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ s.t. $(x,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$. Indeed, consider some $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ and $(x,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$. By the lemma above, $(x_\epsilon,y)(\phi) \in \{\mathsf{T},\mathsf{C}\}$ for every $\epsilon \prec \alpha$. Thus, for every $\epsilon \prec \alpha$, $x_\epsilon \cap \Delta \neq \emptyset$. Since $\{x_\epsilon\}_{\epsilon<\alpha}$ is an infinite descending chain and Δ is finite, there is a $\delta \in \Delta$ s.t. δ is part of an infinite number of sets in $\{x_\epsilon\}_{\epsilon<\alpha}$. Since $\{x_\epsilon\}_{\epsilon<\alpha}$ is a $\subseteq$-descending chain, $\delta \in x_\epsilon$ for every $\epsilon \prec \alpha$, and thus $\delta \in x$.

We can now show that $\mathcal{IC}_{\mathcal{P}}(x,y) \preceq^S_L \{x\}$. Indeed, since $x \cap \Delta \neq \emptyset$ for every $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$, $z = x \cap \bigcup \mathcal{HD}_{\mathcal{P}}(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$ and thus we have found our interpretation $z \in \mathcal{IC}_{\mathcal{P}}(x,y)$ s.t. $z \subseteq x$. □

Theorem 4. Consider a normal disjunctive logic program $\mathcal{P}$ and a consistent interpretation $(x,y) \in \wp(\mathcal{A}_{\mathcal{P}}) \times \wp(\mathcal{A}_{\mathcal{P}})$. Then (x,y) is a stable model of $\mathcal{P}$ iff $(x,y) \in S(\mathcal{IC}_{\mathcal{P}})(x,y)$.

Proof. [$\Rightarrow$] Suppose that (x,y) is a stable model of $\mathcal{P}$. We show that $x \in C(\mathcal{IC}^l_{\mathcal{P}})(y)$. (The proof that $y \in C(\mathcal{IC}^l_{\mathcal{P}})(x)$ (thus $y \in C(\mathcal{IC}^u_{\mathcal{P}})(x)$, since $\mathcal{IC}_{\mathcal{P}}$ is symmetric) is analogous.) We first show that $x \in \mathcal{IC}^l_{\mathcal{P}}(x,y)$. Indeed, this immediately follows from the fact that any stable interpretation is weakly supported and that any weakly supported model is a fixpoint of $\mathcal{IC}_{\mathcal{P}}$ (Theorem 1). It remains to show $\subseteq$-minimality of x among the fixpoints of $\mathcal{IC}^l_{\mathcal{P}}(\cdot,y)$ and of y among the fixpoints of $\mathcal{IC}^u_{\mathcal{P}}(x,\cdot)$. Suppose towards a contradiction that there is some $x' \subset x$ such that $x' \in \mathcal{IC}^l_{\mathcal{P}}(x',y)$. We show that $(x',y) \in mod(\frac{\mathcal{P}}{(x,y)})$, which contradicts $(x,y) \in \min_{\leq_t}(mod(\frac{\mathcal{P}}{(x,y)}))$ (the latter follows from the assumption that (x,y) is stable). Indeed, let $\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \bigwedge_{i=1}^n \neg\beta_i \in \frac{\mathcal{P}}{(x,y)}$. We consider three cases:

- $(x,y)(\neg\beta_i) = \mathsf{T}$ for every $1 \leq i \leq n$. This means that $\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \mathsf{T} \in \frac{\mathcal{P}}{(x,y)}$. Notice that for any $\alpha \in \mathcal{A}_{\mathcal{P}}$, $(x',y)(\alpha) \leq_t (x,y)(\alpha)$ (since $x' \subseteq x$). Now,
  - If $(x',y)(\bigwedge\Theta) = \mathsf{F}$, $(x',y)(\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \mathsf{T})$ is trivially satisfied.
  - If $(x',y)(\bigwedge\Theta) = \mathsf{U}$, $(x',y)(\bigwedge\Theta) \leq_t (x,y)(\bigwedge\Theta) \leq_t (x,y)(\bigvee\Delta)$ implies $(x,y)(\bigwedge\Theta) \in \{\mathsf{T},\mathsf{U}\}$ and thus, since (x,y) is a stable model of $\frac{\mathcal{P}}{(x,y)}$, $(x,y)(\bigvee\Delta) \in \{\mathsf{T},\mathsf{U}\}$, i.e. $\Delta \cap y \neq \emptyset$. Thus, $(x',y)(\bigvee\Delta) \in \{\mathsf{T},\mathsf{U}\}$.
  - If $(x',y)(\bigwedge\Theta) = \mathsf{T}$, then $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x',y)$ (since $x' \in \mathcal{IC}_{\mathcal{P}}(x',y)$), and so (since $(x',y) \in \mathcal{IC}^{\mathsf{cons}}_{\mathcal{P}}(x',y)$), $x' \cap \Delta \neq \emptyset$.
- $(x,y)(\neg\beta_i) \in \{\mathsf{T},\mathsf{U}\}$ for every $1 \leq i \leq n$, and $(x,y)(\neg\beta_i) = \mathsf{U}$ for some $1 \leq i \leq n$. Then $\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \mathsf{U} \in \frac{\mathcal{P}}{(x,y)}$. It can be shown that (x',y) satisfies $\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \mathsf{U}$ just like the previous case.

- $(x,y)(\neg\beta_i) = \mathsf{F}$ for some $1 \leq i \leq n$: trivial.

Altogether, we have shown that (x',y) satisfies any $\bigvee\Delta \leftarrow \bigwedge\Theta \wedge \bigwedge_{i=1}^{n}\neg\beta \in \frac{\mathcal{P}}{(x,y)}$, contradicting the assumption that $(x,y) \in \min_{\leq_t}(mod(\frac{\mathcal{P}}{(x,y)}))$. We conclude then that x is a $\subseteq$-minimal fixpoint of $\mathcal{O}_l(\cdot,y)$. Analogously, it can be shown that y is $\subseteq$-minimal among fixpoints of $\mathcal{IC}^u_{\mathcal{P}}(x,\cdot)$.

[$\Leftarrow$] Suppose now that $(x,y) \in S(\mathcal{IC}_{\mathcal{P}})(x,y)$. We first show that (x,y) is a model of $\frac{\mathcal{P}}{(x,y)}$. Indeed, by Proposition 14 $(x,y) \in \mathcal{IC}_{\mathcal{P}}(x,y)$, thus by Theorem 1 (x,y) is a weakly supported model of $\mathcal{P}$. Since any model (x,y) of $\mathcal{P}$ is also a model of $\frac{\mathcal{P}}{(x,y)}$, we have that (x,y) is a model of $\frac{\mathcal{P}}{(x,y)}$. For $\leq_t$-minimality, suppose towards a contradiction that there is some $(x',y') <_t (x,y)$ such that $(x',y') \in mod(\frac{\mathcal{P}}{(x,y)})$. Since $(x',y') <_t (x,y)$, either $x' \subsetneq x$ or $y' \subsetneq y$. Suppose first that $x' \subsetneq x$. By Lemma 6, $\mathcal{HD}^l_{\mathcal{P}}(x',y) \subseteq \mathcal{HD}^l_{\mathcal{P}}(x,y)$, and for a similar reason $\mathcal{HD}^l_{\mathcal{P}}(x',y) \subseteq \mathcal{HD}^l_{\mathcal{P}}(x',y')$, thus $\mathcal{HD}^l_{\mathcal{P}}(x',y') \subseteq \mathcal{HD}^l_{\mathcal{P}}(x,y)$. We have: $\mathcal{IC}_{\mathcal{P}}(x',y') = \min_{\leq_t}(mod(\frac{\mathcal{P}}{(x',y')}) \subseteq \min_{\leq_t}(mod(\frac{\mathcal{P}}{(x,y)})) = \mathcal{IC}_{\mathcal{P}}(x,y)$. Hence $x' \in \mathcal{IC}_{\mathcal{P}}(x,y)$, but this contradicts the fact that x is a $\leq$-minimal fixpoint of $\mathcal{IC}^l_{\mathcal{P}}(\cdot,y)$ (which follows from the assumption that $(x,y) \in S(\mathcal{IC}_{\mathcal{P}})(x,y)$). The proof of the case where $y' \subsetneq y$ is similar. □

Theorem 7. For any disjunctive logic program $\mathcal{P}$, $\mathsf{WF}(\mathcal{IC}_{\mathcal{P}}) \preceq^A_i \mathsf{WFS}_d(\mathcal{P})$.

Proof. By Proposition 10, $\Gamma_{\mathcal{P}}(X) = \bigcup_{x\in X} C(\mathcal{IC}^l_{\mathcal{P}})(x)$ for any $X \subseteq \wp(\mathcal{A}_{\mathcal{P}})$. Furthermore, as for any $X, Y \subseteq \mathcal{A}_{\mathcal{P}}$, $\mathfrak{T}_Y(X) \subseteq X$, it holds that $\bigcup_{x\in\mathfrak{T}_Y(X)} S(\mathcal{IC}_{\mathcal{P}})(x) \preceq^H_L \bigcup_{x\in X} S(\mathcal{IC}_{\mathcal{P}})(x)$. The proposition now immediately follows from these two observations. □

Appendix B. Additional operator-based characterizations of semantics for disjunctive logic programming

In this appendix, we characterise the supported, regular, M-stable and L-stable model semantics for normal disjunctive logic programs in an operator-based way.

The following represents a three-valued generalization of the supported model semantics introduced by Dix and Brass [15]:

Definition 25. (x,y) is a *supported model* of $\mathcal{P}$, if it is a model of $\mathcal{P}$ and for every $p \in \mathcal{A}_{\mathcal{P}}$ such that $(x,y)(p) = \mathsf{T}$ [$(x,y)(p) = \mathsf{U}$], there is $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ such that $p \in \Delta$ and $(x,y)(\phi) = \mathsf{T}$ [$(x,y)(p) = \mathsf{U}$] and $\Delta \cap x = \{p\}$ [$\Delta \cap y = \{p\}$].

Example 27 *(Example 3 continued).* Consider again the dlp $\mathcal{P} = \{p \vee q \leftarrow q\}$ from Example 3. The following are the supported models of $\mathcal{P}$:

$$(\emptyset,\emptyset), \quad (\emptyset,\{q\}), \quad (\{q\},\{q\}).$$

We turn to the operator-based representation of supported models. As supported models allow for the truth of fewer atoms than the weakly supported models, one might conjecture that supported models are the $\leq_t$-minimal fixpoints of $\mathcal{IC}_{\mathcal{P}}$, but this does not hold, not even for positive or normal programs:

Example 28. Take $\mathcal{P} = \{p \leftarrow r;\ r \leftarrow r\}$. Then $(\{p,r\},\{p,r\})$ and $(\emptyset,\emptyset)$ are both supported models of $\mathcal{P}$, but $(\{p,r\},\{p,r\})$ is not $\leq_t$-minimal.

The supported models can actually be characterized as the fixpoints of $\mathcal{IC}^m_{\mathcal{P}}$ as defined in Remark 7. We refer to our previous work [38] for more details on how this can be done. However, this operator is not $\preceq^A_i$-monotonic (as shown in Remark 7). Next, we show that supported models of a dlp $\mathcal{P}$ may also be characterized by the fixpoints of $\mathcal{IC}_{\mathcal{P}}$, but this time together with $\subseteq$-minimization.

Theorem 8. *Given a dlp $\mathcal{P}$ and a consistent interpretation $(x,y) \in \wp(\mathcal{A}_{\mathcal{P}})^2$. Then (x,y) is a supported model of $\mathcal{P}$ iff $x \in \min_{\subseteq}(\mathcal{IC}^l_{\mathcal{P}}(x,y))$ and $y \in \min_{\subseteq}(\mathcal{IC}^u_{\mathcal{P}}(x,y))$.*

Proof. [$\Rightarrow$] Suppose that (x,y) is a supported model of $\mathcal{P}$ and consider some $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)$. Since (x,y) is in particular weakly supported, by Theorem 1 it follows that $\Delta \cap x \neq \emptyset$. We show that $x \in \min_{\subseteq}(\{v \mid v \cap \Delta \neq \emptyset$ for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)\}$. Indeed, suppose towards a contradiction that there is some $x' \in \{v \mid v \cap \Delta \neq \emptyset$ for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)\}$ such that $x' \subsetneq x$. Let $\alpha \in x \setminus x'$. Then, since (x,y) is supported, there is a $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ such that $\Delta \cap x = \{\alpha\}$ and $(x,y)(\phi) = \mathsf{T}$. But then $x' \notin \mathcal{IC}^l_{\mathcal{P}}(x,y)$, in a contradiction to our assumption that $x' \in \min_{\subseteq}(\mathcal{IC}^l\mathcal{P}(x,y))$. Analogously, we can show that $y \in \min_{\subseteq}(\{v \mid v \cap \Delta \neq \emptyset$ for every $\Delta \in \mathcal{HD}^u_{\mathcal{P}}(x,y)\}$.

[$\Leftarrow$] Suppose that $x \in \min_{\subseteq}(\mathcal{IC}^l_{\mathcal{P}}(x,y))$ and $y \in \min_{\subseteq}(\mathcal{IC}^u_{\mathcal{P}}(x,y))$. By Theorem 1 it follows that (x,y) is weakly supported and thus a model of $\mathcal{P}$.

We now show that (x,y) is supported. Indeed, let $\alpha \in \Delta \cap x$. Suppose first that there is no $\bigvee\Delta \leftarrow \phi \in \mathcal{P}$ s.t. $(x,y)(\phi) = \mathsf{T}$. Then $x \notin min_{\subseteq}(\{v \mid v \cap \Delta \neq \emptyset$ for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)\}$, since there is some $x' \subseteq x \setminus \{\alpha\}$ such that $x' \in min_{\subseteq}(\{v \mid v \cap \Delta \neq \emptyset$ for every $\Delta \in \mathcal{HD}^l_{\mathcal{P}}(x,y)\}$. Similarly for $(x,y)(\phi) = \mathsf{U}$. The proof of the second condition in the definition of supported models is similar. □

Finally, we turn to the characterization of some additional variants of the three-valued stable semantics in our framework, namely the regular [61], M-stable [27] and L-stable semantics [27].

Definition 26. (x, y) is *founded* iff $x \in \min_{\subseteq} mod_2(\frac{\mathcal{P}}{y})$. (x, y) is regular iff (x, y) is a founded model of $\mathcal{P}$ s.t. there is no founded model $(x', y') \leq_i (x, y)$ of $\mathcal{P}$ (i.e. it is a $\leq_i$-maximal founded model of (x', y').[32]

Definition 27. (x, y) is an *M-stable model* of $\mathcal{P}$ iff (x, y) is a three-valued stable model of $\mathcal{P}$ and it is $\leq_i$-maximal among the three-valued stable models of $\mathcal{P}$. (x, y) is an *L-stable model* of $\mathcal{P}$ iff (x, y) is a three-valued stable model of $\mathcal{P}$ and there is no three-valued stable model (x', y') of $\mathcal{P}$ s.t. $y' \setminus x' \subset y \setminus x$.

In what follows we denote by $\mathsf{SF}(\mathcal{IC}_{\mathcal{P}})$ the stable fixpoints of $\mathcal{IC}_{\mathcal{P}}$.

Theorem 9. *Let a disjunctively normal logic program $\mathcal{P}$ be given.*

1. *A three-valued interpretation (x, y) is founded iff $x \in C(\mathcal{IC}^u_{\mathcal{P}})(y)$.*
2. *A three-valued interpretation (x, y) is regular iff $x \in C(\mathcal{IC}^u_{\mathcal{P}})(y)$, $\mathcal{IC}_{\mathcal{P}}(x, y) \preceq^S_t (x, y)$ and it is $\leq_i$-maximal with these properties.*
3. *A three-valued interpretation (x, y) is an M-stable model iff $(x, y) \in \max_{\leq_i}(\mathsf{SF}(\mathcal{IC}_{\mathcal{P}}))$.*
4. *A three-valued interpretation (x, y) is an L-stable model iff $(x, y) \in \arg_{(x,y)\in \mathsf{SF}(\mathcal{IC}_{\mathcal{P}})} \min_{\subseteq}(y \setminus x)$.*

Proof. Item 1. By Proposition 10, $C(\mathcal{IC}_{\mathcal{P}})(y) = \min_{\subseteq}(mod_2(\frac{\mathcal{P}}{y}))$ for any $y \subseteq \mathcal{A}_{\mathcal{P}}$, which suffices to show the claim.

Item 2. We first note that in the proof of Theorem 1, it is shown that $\mathcal{IC}_{\mathcal{P}}(x, y) \preceq^S_t (x, y)$ iff (x, y) is a model of $\mathcal{P}$. Thus, (x, y) is a founded model of $\mathcal{P}$ iff $x \in C(\mathcal{IC}^u_{\mathcal{P}})(y)$ and $\mathcal{IC}_{\mathcal{P}}(x, y) \preceq^S_t (x, y)$, which suffices to show the claim.

Items 3 and 4. Immediate from Theorem 4. □

References

[1] Felicidad Aguado, Pedro Cabalar, Jorge Fandinno, David Pearce, Gilberto Pérez, Concepción Vidal, Forgetting auxiliary atoms in forks, Artif. Intell. 275 (2019) 575–601.
[2] Joao Alcântara, Carlos Viegas Damásio, Luís Moniz Pereira, A well-founded semantics with disjunction, in: Proceedings of ICLP'05, Springer, 2005, pp. 341–355.
[3] Christian Antić, Thomas Eiter, Michael Fink, Hex semantics via approximation fixpoint theory, in: Proceedings of LPNMR'13, in: LNCS, vol. 8148, Springer, 2013, pp. 102–115.
[4] Arieli Ofer, Arnon Avron, Reasoning with logical bilattices, J. Log. Lang. Inf. 5 (1) (1996) 25–63.
[5] Mathieu Beirlaen, Jesse Heyninck, Christian Straßer, Reasoning by cases in structured argumentation, in: Proceedings of SAC'17, 2017, pp. 989–994.
[6] Mathieu Beirlaen, Jesse Heyninck, Christian Straßer, A critical assessment of pollock's work on logic-based argumentation with suppositions, in: Proceedings of NMR'18, vol. 20, 2018, pp. 63–72.
[7] Nuel D. Belnap, How a computer should think, in: G. Ryle (Ed.), Contemporary Aspects of Philosophy, Oriel Press, 1977.
[8] Nuel D. Belnap, A useful four-valued logic, in: J.M. Dunn, G. Epstein (Eds.), Modern Uses of Multiple-Valued Logics, Reidel Publishing Company, 1977, pp. 7–37.
[9] Bart Bogaerts, Weighted abstract dialectical frameworks through the lens of approximation fixpoint theory, in: Proceedings of AAAI'19, vol. 33, 2019, pp. 2686–2693.
[10] Bart Bogaerts, Luís Cruz-Filipe, Fixpoint semantics for active integrity constraints, Artif. Intell. 255 (2018) 43–70.
[11] Bart Bogaerts, Maxime Jakubowski, Fixpoint semantics for recursive SHACL, in: Proceedings 37th International Conference on Logic Programming (Technical Communications), ICLP Technical Communications 2021, 2021, pp. 41–47.
[12] Bart Bogaerts, Joost Vennekens, Marc Denecker, Grounded fixpoints and their applications in knowledge representation, Artif. Intell. 224 (2015) 51–71.
[13] Bart Bogaerts, Joost Vennekens, Marc Denecker, Safe inductions and their applications in knowledge representation, Artif. Intell. 259 (2018) 167–185.
[14] Daniel Bonevac, Defaulting on reasons, Noûs 52 (2) (2018) 229–259.
[15] Stefan Brass, Jürgen Dix, Characterizations of the stable semantics by partial evaluation, in: Proceedings of LPNMR'95, Springer, 1995, pp. 85–98.
[16] Pedro Cabalar, Sergei Odintsov, David Pearce, Agustín Valverde, Analysing and extending well-founded and partial stable semantics using partial equilibrium logic, in: Proceedings of ICLP'06, Springer, 2006, pp. 346–360.
[17] Pedro Cabalar, Sergei Odintsov, David Pearce, Agustín Valverde, Partial equilibrium logic, Ann. Math. Artif. Intell. 50 (3) (2007) 305–331.
[18] Angelos Charalambidis, Panos Rondogiannis, Ioanna Symeonidou, Approximation fixpoint theory and the well-founded semantics of higher-order logic programs, Theory Pract. Log. Program. 18 (3–4) (2018) 421–437.
[19] Patrick Cousot, Radhia Cousot, Constructive versions of Tarski's fixed point theorems, Pac. J. Math. 82 (1) (1979) 43–57.
[20] Ingmar Dasseville, Matthias van der Hallen, Bart Bogaerts, Gerda Janssens, Marc Denecker, A compositional typed higher-order logic with definitions, in: Technical Communications of the 32nd International Conference on Logic Programming, ICLP, 2016, pp. 14:1–14:13.
[21] Marc Denecker, Maurice Bruynooghe, Joost Vennekens, Approximation fixpoint theory and the semantics of logic and answers set programs, in: Correct Reasoning, Springer, 2012, pp. 178–194.
[22] Marc Denecker, V. Wiktor Marek, Miroslaw Truszczynski, Ultimate approximations in nonmonotonic knowledge representation systems, in: Proceedings of KR'02, 2002, pp. 177–190.
[23] Marc Denecker, Victor Marek, Mirosław Truszczyński, Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning, in: Logic-Based Artificial Intelligence, in: Engineering and Computer Science, vol. 597, Springer, 2000, pp. 127–144.
[24] Marc Denecker, Victor Marek, Mirosław Truszczyński, Uniform semantic treatment of default and autoepistemic logics, Artif. Intell. 143 (1) (2003) 79–122.
[25] Phan Minh Dung, Paolo Mancarella, Francesca Toni, Computing ideal sceptical argumentation, Artif. Intell. 171 (10–15) (2007) 642–674.

[32] This formulation is an adaptation to our setting of the definition by Eiter et al. [27] shown to be equivalent to the original definition of regular models [61].

[26] Thomas Eiter, Georg Gottlob, Complexity results for disjunctive logic programming and application to nonmonotonic logics, in: Proceedings of ILPS'93, 1993, pp. 266–278.
[27] Thomas Eiter, Nicola Leone, Domenico Sacca, On the partial semantics for disjunctive deductive databases, Ann. Math. Artif. Intell. 19 (1–2) (1997) 59–96.
[28] José Alberto Fernández, Jack Minker, Bottom-up computation of perfect models for disjunctive theories, J. Log. Program. 25 (1) (1995) 33–51.
[29] Melvin Fitting, Bilattices and the semantics of logic programming, J. Log. Program. 11 (2) (1991) 91–116.
[30] Melvin Fitting, Bilattices are nice things, in: Self Reference, in: CSLI Lecture Notes, vol. 178, CLSI Publications, 2006, pp. 53–77.
[31] Melvin Fitting, Bilattice basics, IfCoLog J. Log. Appl. 7 (6) (2020) 973–1016.
[32] Michael Gelfond, Vladimir Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (3–4) (1991) 365–385.
[33] Michael Gelfond, Halina Przymusinska, Vladimir Lifschitz, Miroslaw Truszczynski, Disjunctive defaults, in: Proceedings of KR'91, 1991, pp. 230–237.
[34] Matthew L. Ginsberg, Multi-valued logics: a uniform approach to reasoning in artificial intelligence, Comput. Intell. 4 (1988) 256–316.
[35] Pieter Van Hertum, Marcos Cramer, Bart Bogaerts, Marc Denecker, Distributed autoepistemic logic and its application to access control, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI, 2016, pp. 1286–1292.
[36] Jesse Heyninck, Semantics for logic programs with choice constructs on the basis of approximation fixpoint theory (preliminary report), in: 21st International Workshop on Non-Monotonic Reasoning, NMR 2023, CEUR-WS, 2023, pp. 74–83.
[37] Jesse Heyninck, Ofer Arieli, Argumentative reflections of approximation fixpoint theory, in: Computational Models of Argument: Proceedings of COMMA 2020, vol. 326, 2020, p. 215.
[38] Jesse Heyninck, Ofer Arieli, Approximation fixpoint theory for non-deterministic operators and its application in disjunctive logic programming, in: Proceedings of KR'21, 2021, pp. 334–344.
[39] Jesse Heyninck, Bart Bogaerts, Non-deterministic approximation operators: Ultimate operators, semi-equilibrium semantics, and aggregates, Theory Pract. Log. Program. 23 (4) (2023) 632–647.
[40] Jesse Heyninck, Matthias Thimm, Gabriele Kern-Isberner, Tjitze Rienstra, Kenneth Skiba, Conditional abstract dialectical frameworks, in: Proceedings of AAAI'22, AAAI Press, 2022, pp. 5692–5699.
[41] Kazimierz Kuratowski, Une méthode d'élimination des nombres transfinis des raisonnements mathématiques, Fundam. Math. 3 (1) (1922) 76–108.
[42] Fangfang Liu, Jia-Huai You, Alternating fixpoint operator for hybrid MKNF knowledge bases as an approximator of AFT, Theory Pract. Log. Program. 22 (2) (2022) 305–334.
[43] Jorge Lobo, Jack Minker, Arcot Rajasekar, Foundations of Disjunctive Logic Programming, MIT Press, 1992.
[44] Jack Minker, Dietmar Seipel, Disjunctive logic programming: a survey and assessment, in: Computational Logic: Logic Programming and Beyond, Springer, 2002, pp. 472–511.
[45] Søren Holbech Nielsen, Simon Parsons, A generalization of Dung's abstract framework for argumentation: arguing with sets of attacking arguments, in: Proceedings of ArgMAS'06, Springer, 2006, pp. 54–73.
[46] David Pearce, Equilibrium logic, Ann. Math. Artif. Intell. 47 (1) (2006) 3–41.
[47] Nikolay Pelov, Marc Denecker, Maurice Bruynooghe, Well-founded and stable semantics of logic programs with aggregates, Theory Pract. Log. Program. 7 (3) (2007) 301–353.
[48] Nikolay Pelov, Miroslaw Truszczynski, Semantics of disjunctive programs with monotone aggregates: an operator-based approach, in: Proceedings of NMR'04, 2004, pp. 327–334.
[49] Teodor C. Przymusinski, The well-founded semantics coincides with the three-valued stable semantics, Fundam. Inform. 13 (4) (1990) 445–463.
[50] Chiaki Sakama, Katsumi Inoue, Paraconsistent stable semantics for extended disjunctive programs, J. Log. Comput. 5 (3) (1995) 265–285.
[51] Dietmar Seipel, An alternating well-founded semantics for query answering in disjunctive databases, in: Proceedings of FQAS'98, Springer, 1998, pp. 341–353.
[52] Michael B. Smyth, Powerdomains, in: Proceedings of MFCS'76, Springer, 1976, pp. 537–543.
[53] Hannes Strass, Approximating operators and semantics for abstract dialectical frameworks, Artif. Intell. 205 (2013) 39–70.
[54] Hannes Strass, Johannes Peter Wallner, Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory, Artif. Intell. 226 (2015) 34–74.
[55] Alfred Tarski, On the concept of logical consequence, in: Logic, Semantics, Metamathematics, vol. 52, 1936, pp. 409–420.
[56] Mirosław Truszczyński, Strong and uniform equivalence of nonmonotonic theories–an algebraic approach, Ann. Math. Artif. Intell. 48 (3) (2006) 245–265.
[57] Miroslaw Truszczyński, Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs, Artif. Intell. 174 (16–17) (2010) 1285–1306.
[58] Maarten H. van Emden, Robert A. Kowalski, The semantics of predicate logic as a programming language, J. ACM 23 (4) (1976) 733–742.
[59] Allen Van Gelder, Kenneth A. Ross, John S. Schlipf, The well-founded semantics for general logic programs, J. ACM 38 (3) (1991) 619–649.
[60] Joost Vennekens, David Gilis, Marc Denecker, Splitting an operator: algebraic modularity results for logics with fixpoint semantics, ACM Trans. Comput. Log. 7 (4) (2006) 765–797.
[61] Jia-Huai You, Li Yan Yuan, A three-valued semantics for deductive databases and logic programs, J. Comput. Syst. Sci. 49 (2) (1994) 334–361.
[62] Max Zorn, A remark on method in transfinite algebra, Bull. Am. Math. Soc. 41 (10) (1935) 667–670.