# Epistemic foundation of stable model semantics

## YANN LOYER

*Laboratoire PRiSM, Université de Versailles Saint Quentin, Versailles, France*
(*e-mail:* `Yann.Loyer@prism.uvsq.fr`)

## UMBERTO STRACCIA

*Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche, Pisa, Italy*
(*e-mail:* `straccia@isti.cnr.it`)

## Abstract

*Stable model semantics* has become a very popular approach for the management of negation in logic programming. This approach relies mainly on the closed world assumption to complete the available knowledge and its formulation has its basis in the so-called Gelfond–Lifschitz transformation. The primary goal of this work is to present an alternative and epistemic-based characterization of stable model semantics, to the Gelfond-Lifschitz transformation. In particular, we show that stable model semantics can be defined entirely as an extension of the Kripke-Kleene semantics. Indeed, we show that the closed world assumption can be seen as an additional source of 'falsehood' to be added *cumulatively* to the Kripke-Kleene semantics. Our approach is purely algebraic and can abstract from the particular formalism of choice as it is based on monotone operators (under the knowledge order) over bilattices only.

## 1 Introduction

*Stable model semantics* (Gelfond and Lifschitz 1988; Gelfond and Lifschitz 1991) is probably the most widely studied and most commonly accepted approach adopted to give meaning to logic programs (with negation). Informally, it consists in relying on the *Closed World Assumption* (CWA) to *complete* the available knowledge –CWA assumes that all atoms not entailed by a program are false (see Reiter (1978)), and is motivated by the fact that explicit representation of negative information in logic programs is not feasible since the addition of explicit negative information could overwhelm a system. Defining default rules which allow implicit inference of negated facts from positive information encoded in a logic program has been an attractive alternative to the explicit representation approach.

Stable model semantics defines a whole family of models of (or 'answers to') a logic program and, remarkably, one of these stable models, the *minimal* one according

to 'knowledge or information ordering', is taken as the favorite (Denecker 1998; Denecker *et al.* 2001; Przymusinski 1990c), and is one-to-one related with the so-called *well-founded semantics* (van Gelder 1989; van Gelder *et al.* 1991).

The original formulation of stable model semantics was classical, two-valued, over the set of truth-values $\{f, t\}$. But, some programs have no stable model under this setting. To overcome this problem, Przymusinski (1990a; 1990b; 1990c) extended the notion of stable model semantics to allow three-valued, or partial, stable model semantics. Remarkably, three-valued logics has also been considered in other approaches for providing semantics to logic programs, such as in Fitting (1985) and Kunen (1987), where Clark's completion is extended to three-valued logics, yielding the well-known *Kripke-Kleene semantics* of logic programs. In three-valued semantics, the set of truth values is $\{f, t, \perp\}$, where $\perp$ stands for *unknown*. Przymusinski showed that every program has at least a partial stable model and that the well-founded model is the smallest among them, according to the knowledge ordering. It is then a natural step to move from a three-valued semantics, allowing the representation of incomplete information, to a four-valued semantics, allowing the representation of *inconsistency* (denoted $\top$). The resulting semantics is based on the well-known set of truth-values $\mathscr{FOUR} = \{f, t, \perp, \top\}$, introduced by Belnap (1977) to model a kind of 'relevance logic' (there should be some 'syntactical' connections between the antecedent and the consequent of a logical entailment relation $\alpha \models \beta$ (Anderson and Belnap 1975; Dunn 1976; Dunn 1986; Levesque 1984; Levesque 1988). This process of enlarging the set of truth-values culminated with Fitting's progressive work (1985; 1991; 1992; 1993; 2002) on giving meaning to logic programs by relying on *bilattices* (Ginsberg 1988). Bilattices, where $\mathscr{FOUR}$ is the simplest non-trivial one, play an important role in logic programming, and in knowledge representation in general. Indeed, Arieli and Avron (1996; 1998) show that the use of four values is preferable to the use of three values even for tasks that can in principle be handled using only three values. Moreover, Fitting explains clearly (Fitting 1991) why $\mathscr{FOUR}$ can be thought of as the 'home' of classical logic programming. Interestingly, the algebraic work of Fitting's fixed-point characterisation of stable model semantics on bilattices (Fitting 1993; Fitting 2002) has been the starting point of the work carried out by Denecker, Marek and Truszczyński (1999; 2002; 2003), who extended Fitting's work to a more abstract context of fixed-points operators on lattices, by relying on *interval* bilattices (these bilattices are obtained in a standard way as a product of a lattice – see, for instance, Fitting (1993, 1992). Denecker, Marek and Truszczyński (1999; 2003) showed interesting connections between (two-valued and four-valued) Kripke-Kleene (Fitting 1985), well-founded and stable model semantics, as well as to Moore's (1984) autoepistemic logic and Reiter's (1980) default logic. Other well-established applications of bilattices and/or Kripke-Kleene, well-founded and stable model semantics to give semantics to logic programs can be found in the context of reasoning under paraconsistency and uncertainty (Alcântâra *et al.* 2002; Arieli 2002; Blair and Subrahmanian 1989; Damásio and Pereira 1998; Damásio and Pereira 2001; Loyer and Straccia 2002a; Loyer and Straccia 2002b; Loyer and Straccia 2003a; Loyer and Straccia 2003b; Lukasiewicz 2001; Ng and Subrahmanian 1991). Technically, classical two-valued

stable models of logic programs are defined in terms of fixed-points of the so-called Gelfond-Lifschitz operator, $GL(I)$, for a two-valued interpretation $I$. This operator has been generalized to bilattices by Fitting (1993), by means of the $\Psi'_{\mathscr{P}}(I)$ operator, where this time $I$ is an interpretation over bilattices. Informally, the main principle of these operators is based on the *separation of the role of positive and negative information*. That is, given a two-valued interpretation $I$, $GL(I)$ is obtained by first evaluating negative literals in a logic program $\mathscr{P}$ by means of $I$, determining the *reduct* $\mathscr{P}^I$ of $\mathscr{P}$, and then, as $\mathscr{P}^I$ is now a positive program, to compute the minimal Herbrand model of $\mathscr{P}^I$ by means of the usual Van Emden-Kowalski's immediate consequence operator $T_{\mathscr{P}}$ (Emden and Kowalski 1976; Lloyd 1987). The computation of $\Psi'_{\mathscr{P}}(I)$ for bilattices is similar. As a consequence, this separation avoids the natural management of classical negation (i.e. the evaluation of a negative literal $\neg A$ is given by the negation of the evaluation of $A$), which is a major feature of the Kripke-Kleene semantics (Fitting 1985; Fitting 1991) of logic programs with negation.

The primary goal of this study is to show, in the quite general setting of bilattices as space of truth-values, that this separation of positive and negative information is nor necessary nor is any program transformation required to characterize stable model semantics epistemologically. Another motivation is to evidence the role of CWA as a discriminating factor between the most commonly accepted semantics of logic programs. We show that the only difference between Kripke-Kleene, well-founded and stable model semantics is the amount of knowledge taken from CWA that they integrate. We view CWA, informally as an additional source of information to be used for information completion, or more precisely, as a carrier of falsehood, to be considered cumulatively to Kripke-Kleene semantics. This allows us to view stable model semantics from a different, not yet investigated perspective. Roughly speaking, in Kripke-Kleene semantics, CWA is used to consider only those atoms that do not appear in head of any rule as false (and that can obviously not be inferred as true), while the well-founded and stable model semantics integrate more CWA-provided knowledge. To identify this knowledge, we introduce the notion of *support*. This is a generalization of the notion of *greatest unfounded set* (van Gelder *et al.* 1991) (which determines the atoms that can be assumed to be false) to the bilattice context. It determines in a principled way the amount of falsehood provided by CWA that can "safely" be assumed. More precisely, as we are considering a many-valued truth space, it provides the degree of falseness that can "safely" be assumed for each atom. We then show how the support can be used to complete the Kripke-Kleene semantics in order to obtain the well-founded and stable model semantics over bilattices. In particular, we show that the well-founded semantics is the least informative model in the set of models containing their own support, while a model is a stable model if and only if it is deductively closed under support completion, i.e. it contains exactly the knowledge that can be inferred by activating the rules over the support. We thus show an alternative characterisation of the stable model semantics to the well-known, widely applied and long studied technique based on the separation of positive and negative information in the Gelfond-Lifschitz transformation, by reverting to the classical interpretation of negation. While the

Gelfond-Lifschitz transformation treats negation-as-failure in a special way and unlike other connectives, our approach is an attempt to relate the semantics of logic programs to a standard model-theoretic account of rules. We show that logic programs can be analyzed using standard logical means such as the notion of interpretation and information ordering, i.e. knowledge ordering. Therefore, in principle, our approach does not depend on the presence of any specific connective, such as negation-as-failure, nor on any specific rule syntax (the work of Herre and Wagner (1997), is in this direction, even if it differs slightly from the usual stable model semantics (Gelfond and Lifschitz 1991) and the semantics is given in the context of the classical, two-valued truth-space). Due to the generality and the purely algebraic nature of our results, as just monotone operators over bilattices are postulated, the epistemic characterisation of stable models given in this study can also be applied in other contexts (e.g. uncertainty and/or paraconsistency in logic programming, and nonmonotonic logics such as default and autoepistemic logics).

The rest of the paper is organized as follows. To make the paper self-contained, in the next section, we will provide definitions and properties of bilattices and logic programs. Section 3 is the main part of this work, where we present our characterisation of the stable model semantics, while Section 4 concludes.

## 2 Preliminaries

### 2.1 Lattices

A *lattice* is a partially ordered set $\langle L, \preceq \rangle$ such that every two element set $\{x, y\} \subseteq L$ has a *least upper bound*, $lub_{\preceq}(x, y)$ (called the *join* of $x$ and $y$), and a *greatest lower bound*, $glb_{\preceq}(x, y)$ (called the *meet* of $x$ and $y$). For ease, we will write $x \prec y$ if $x \preceq y$ and $x \neq y$. A lattice $\langle L, \preceq \rangle$ is *complete* if every subset of $L$ has both least upper and greatest lower bounds. Consequently, a complete lattice has a least element, $\perp$, and a greatest element $\top$. For ease, throughout the paper, given a complete lattice $\langle L, \preceq \rangle$ and a subset of elements $S \subseteq L$, with $\preceq$-*least* and $\preceq$-*greatest* we will always mean $glb_{\preceq}(S)$ and $lub_{\preceq}(S)$, respectively. With $\min_{\preceq}(S)$ we denote the set of minimal elements in $S$ w.r.t. $\preceq$, i.e. $\min_{\preceq}(S) = \{x \in S : \nexists y \in S \text{ s.t. } y \prec x\}$. Note that while $glb_{\preceq}(S)$ is unique, $|\min_{\preceq}(S)| > 1$ may hold. If $\min_{\preceq}(S)$ is a singleton $\{x\}$, for convenience we may also write $x = \min_{\preceq}(S)$ in place of $\{x\} = \min_{\preceq}(S)$. An *operator* on a lattice $\langle L, \preceq \rangle$ is a function from $L$ to $L$, $f: L \to L$. An operator $f$ on $L$ is *monotone*, if for every pair of elements $x, y \in L$, $x \preceq y$ implies $f(x) \preceq f(y)$, while $f$ is *antitone* if $x \preceq y$ implies $f(y) \preceq f(x)$. A *fixed-point* of $f$ is an element $x \in L$ such that $f(x) = x$.

The basic tool for studying fixed-points of operators on lattices is the well-known Knaster-Tarski theorem (Tarski 1955).

*Theorem 2.1* (*Knaster-Tarski fixed-point theorem (Tarski 1955)*)
Let $f$ be a monotone operator on a complete lattice $\langle L, \preceq \rangle$. Then $f$ has a fixed-point, the set of fixed-points of $f$ is a complete lattice and, thus, $f$ has a $\preceq$-*least* and a $\preceq$-*greatest* fixed-point. The $\preceq$-*least* (respectively, $\preceq$-*greatest*) fixed-point can

be obtained by iterating $f$ over $\bot$ (respectively, $\top$), i.e. is the limit of the non-decreasing (respectively, non-increasing) sequence $x_0, \ldots, x_i, x_{i+1}, \ldots, x_\lambda, \ldots$, where for a successor ordinal $i \geqslant 0$,

$$
\begin{aligned}
x_0 &= \bot, \\
x_{i+1} &= f(x_i)
\end{aligned}
$$

(respectively, $x_0 = \top$), while for a limit ordinal $\lambda$,

$$
x_\lambda = lub_{\leq}\{x_i : i < \lambda\} \ (respectively, x_\lambda = glb_{\leq}\{x_i : i < \lambda\}). \tag{1}
$$

We denote the $\leq$-least and the $\leq$-greatest fixed-point by $\mathrm{lfp}_{\leq}(f)$ and $\mathrm{gfp}_{\leq}(f)$, respectively.

Throughout the paper, we will frequently define monotone operators, whose sets of fixed-points define certain classes of models of a logic program. As a consequence, please note that this also means that a least model *always* exists for such classes. Additionally, for ease, for the monotone operators defined in this study, we will specify the initial condition $x_0$ and the next iteration step $x_{i+1}$ only, while Equation (1) is always considered as implicit. To prove that a property holds for a limit ordinal of an iterated sequence, i.e. for transfinite induction, one usually relies on a routine least upper bound (or greatest lower bound) argument and on the Knaster-Tarski theorem. Therefore that case will be considered only in the proof of Theorem 3.10, while the reasoning is similar for all the other proofs and, thus, will be omitted.

### 2.2 Bilattices

The simplest non-trivial bilattice, called $\mathscr{FOUR}$, was defined by Belnap (1977) (see also Arieli and Avron (1998), Avron (1996) and Ginsberg (1988)), who introduced a logic intended to deal with incomplete and/or inconsistent information. $\mathscr{FOUR}$ already illustrates many of the basic properties of bilattices. Essentially, it extends the classical truth set $\{\mathtt{f}, \mathtt{t}\}$ to its power set $\{\{\mathtt{f}\}, \{\mathtt{t}\}, \emptyset, \{\mathtt{f}, \mathtt{t}\}\}$, where we can think that each set indicates the amount of information we have in terms of truth: so, $\{\mathtt{f}\}$ stands for *false*, $\{\mathtt{t}\}$ for *true* and, quite naturally, $\emptyset$ for lack of information or *unknown*, and $\{\mathtt{f}, \mathtt{t}\}$ for *inconsistent* information (for ease, we use $\mathtt{f}$ for $\{\mathtt{f}\}$, $\mathtt{t}$ for $\{\mathtt{t}\}$, $\bot$ for $\emptyset$ and $\top$ for $\{\mathtt{f}, \mathtt{t}\}$). The set of truth values $\{\mathtt{f}, \mathtt{t}, \bot, \top\}$ has two quite intuitive and natural 'orthogonal' orderings, $\leq_k$ and $\leq_t$ (see Figure 1), each giving to $\mathscr{FOUR}$ the structure of a complete lattice. One is the so-called *knowledge ordering*, denoted $\leq_k$, and is based on the subset relation, that is, if $x \subseteq y$ then $y$ represents 'more information' than $x$ (e.g. $\bot = \emptyset \subseteq \{\mathtt{t}\} = \mathtt{t}$, i.e. $\bot \leq_k \mathtt{t}$). The other ordering is the so-called *truth ordering*, denoted $\leq_t$. Here $x \leq_t y$ means that $x$ is 'at least as false as $y$, and $y$ is at least as true as $x$', i.e. $x \cap \{\mathtt{t}\} \subseteq y \cap \{\mathtt{t}\}$ and $y \cap \{\mathtt{f}\} \subseteq x \cap \{\mathtt{f}\}$ (e.g. $\bot \leq_t \mathtt{t}$).

The general notion of bilattice used in this paper is defined as follows (Fitting 2002; Ginsberg 1988). A *bilattice* is a structure $\langle \mathscr{B}, \leq_t, \leq_k \rangle$ where $\mathscr{B}$ is a non-empty set and $\leq_t$ and $\leq_k$ are both partial orderings giving $\mathscr{B}$ the structure of a *complete*
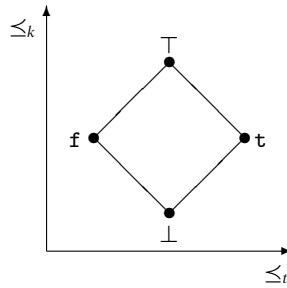
Fig. 1. The logic $\mathscr{FOUR}$.

*lattice* with a top and bottom element. *Meet and join under* $\leq_t$, denoted $\wedge$ and $\vee$, correspond to extensions of classical conjunction and disjunction. On the other hand, *meet and join under* $\leq_k$ are denoted $\otimes$ and $\oplus$. $x \otimes y$ corresponds to the maximal information $x$ and $y$ can agree on, while $x \oplus y$ simply combines the information represented by $x$ with that represented by $y$. *Top and bottom under* $\leq_t$ are denoted t and f, and *top and bottom under* $\leq_k$ are denoted $\top$ and $\bot$, respectively. We will assume that bilattices are *infinitary distributive bilattices* in which all distributive laws connecting $\wedge, \vee, \otimes$ and $\oplus$ hold. We also assume that every bilattice satisfies the *infinitary interlacing conditions*, i.e. each of the lattice operations $\wedge, \vee, \otimes$ and $\oplus$ is monotone w.r.t. both orderings. An example of interlacing condition is: $x \leq_t y$ and $x' \leq_t y'$ implies $x \otimes x' \leq_t y \otimes y'$. Finally, we assume that each bilattice has a *negation*, i.e. an operator $\neg$ that reverses the $\leq_t$ ordering, leaves unchanged the $\leq_k$ ordering, and verifies $\neg\neg x = x$ [1].

Below, we give some properties of bilattices that will be used in this study. Figure 2 illustrates intuitively some of the following lemmas.

*Lemma 2.2 (Fitting 1993)*
  1. If $x \leq_t y \leq_t z$ then $x \otimes z \leq_k y$ and $y \leq_k x \oplus z$;
  2. If $x \leq_k y \leq_k z$ then $x \wedge z \leq_t y$ and $y \leq_t x \vee z$.

*Lemma 2.3*
If $x \leq_t y$ then $x \leq_t x \otimes y \leq_t y$ and $x \leq_t x \oplus y \leq_t y$.

*Proof*
Straightforward using the interlacing conditions.   □

*Lemma 2.4*
  1. If $x \leq_t y$ then $f \otimes x \leq_t y$;
  2. If $x \leq_k y$ then $f \otimes y \leq_t x$.

*Proof*
If $x \leq_t y$ then from $f \leq_t x$ and by Lemma 2.3, $f \leq_t f \otimes x \leq_t x \leq_t y$. If $x \leq_k y$ then, from $f \leq_t x$, we have $f \otimes y \leq_t x \otimes y = x$.   □

---

[1] The dual operation to negation is *conflation*, i.e. an operator $\sim$ that reverses the $\leq_k$ ordering, leaves unchanged the $\leq_t$ ordering, and $\sim\sim x = x$. If a bilattice has both, they *commute* if $\sim \neg x = \neg \sim x$ for all $x$. We will not deal with conflation in this paper.
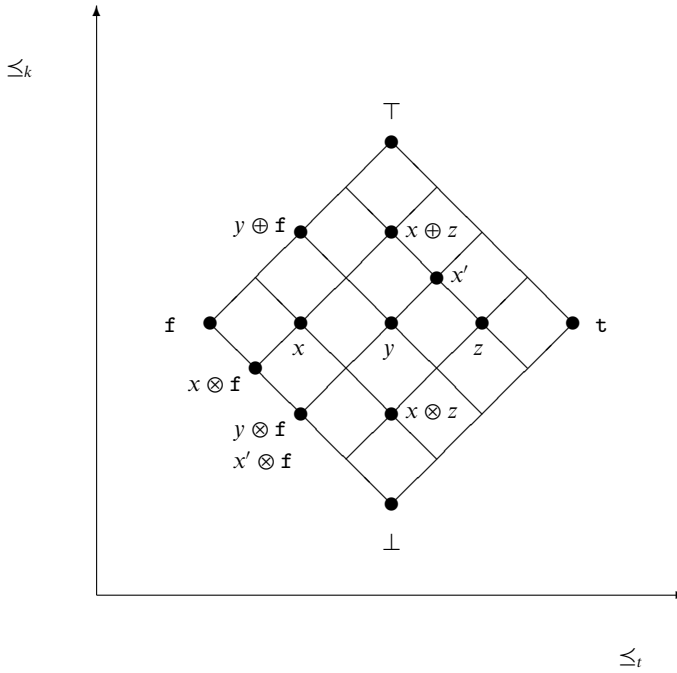
Fig. 2. Some points mentioned in Lemmas 2.2–2.7.

*Lemma 2.5*
If $x \oplus z \preceq_t y$ then $z \preceq_k y \oplus \mathtt{f}$.

*Proof*
By Lemma 2.2, $\mathtt{f} \preceq_t x \oplus z \preceq_t y$ implies $z \preceq_k x \oplus z \preceq_k y \oplus \mathtt{f}$.  □

*Lemma 2.6*
If $\mathtt{f} \otimes y \preceq_k x \preceq_k \mathtt{f} \oplus y$ then $x \preceq_t y$.

*Proof*
By Lemma 2.2, $\mathtt{f} \otimes y \preceq_k x \preceq_k \mathtt{f} \oplus y$ implies $x \preceq_t (\mathtt{f} \otimes y) \vee (\mathtt{f} \oplus y)$. Therefore,
$x \preceq_t (\mathtt{f} \otimes y) \oplus ((\mathtt{f} \otimes y) \vee y)$ and, thus, $x \preceq_t (\mathtt{f} \otimes y) \oplus y = y$.  □

*Lemma 2.7*
If $x \preceq_k y$ and $x \preceq_t y$ then $x \otimes \mathtt{f} = y \otimes \mathtt{f}$.

*Proof*
By Lemma 2.4, $\mathtt{f} \otimes y \preceq_t x$ and, thus, $\mathtt{f} \otimes y \preceq_t x \otimes \mathtt{f}$ follows. From $x \preceq_t y$, $\mathtt{f} \otimes x \preceq_t y \otimes \mathtt{f}$
holds. Therefore, $x \otimes \mathtt{f} = y \otimes \mathtt{f}$.  □

### 2.2.1 Bilattice construction

Bilattices come up in natural ways. There are two general, but different, construction
methods, to build a bilattice from a lattice which are widely used. We only outline
them here in order to give an idea of their application (see also Fitting (1993) and
Ginsberg (1988)).

The first bilattice construction method was proposed by Ginsberg (1988). Suppose we have two complete distributive lattices $\langle L_1, \preceq_1 \rangle$ and $\langle L_2, \preceq_2 \rangle$. Think of $L_1$ as a lattice of values we use when we measure the degree of belief, while think of $L_2$ as the lattice we use when we measure the degree of doubt. Now, we define the structure $L_1 \odot L_2$ as follows. The structure is $\langle L_1 \times L_2, \preceq_t, \preceq_k \rangle$, where

- $\langle x_1, x_2 \rangle \preceq_t \langle y_1, y_2 \rangle$ if $x_1 \preceq_1 y_1$ and $y_2 \preceq_2 x_2$;
- $\langle x_1, x_2 \rangle \preceq_k \langle y_1, y_2 \rangle$ if $x_1 \preceq_1 y_1$ and $x_2 \preceq_2 y_2$.

In $L_1 \odot L_2$ the idea is: knowledge goes up if both degree of belief and degree of doubt go up; truth goes up if the degree of belief goes up, while the degree of doubt goes down. It can easily be verified that $L_1 \odot L_2$ is a bilattice. Furthermore, if $L_1 = L_2 = L$, i.e. we are measuring belief and doubt in the same way (e.g. $L = \{\mathtt{f}, \mathtt{t}\}$), then negation can be defined as $\neg\langle x, y \rangle = \langle y, x \rangle$, i.e. negation switches the roles of belief and doubt. Applications of this method can be found elsewhere (Alcântâra *et al.* 2002; Ginsberg 1988; Herre and Wagner 1997).

The second construction method has been sketched in Ginsberg (1988) and addressed in more detail in Fitting (1992), and is probably the more used one. Suppose we have a complete distributive lattice of truth values $\langle L, \preceq \rangle$. Think of these values as the 'real' values in which we are interested, but due to lack of knowledge we are able just to 'approximate' the exact values. Rather than considering a pair $\langle x, y \rangle \in L \times L$ as indicator for degree of belief and doubt, $\langle x, y \rangle$ is interpreted as the set of elements $z \in L$ such that $x \preceq z \preceq y$. That is, a pair $\langle x, y \rangle$ is interpreted as an *interval*. An interval $\langle x, y \rangle$ may be seen as an approximation of an exact value. For instance, in reasoning under uncertainty (Loyer and Straccia 2002b; Loyer and Straccia 2003a; Loyer and Straccia 2003b), $L$ is the unit interval $[0, 1]$ with standard ordering, $L \times L$ is interpreted as the set of (closed) intervals in $[0, 1]$, and the pair $\langle x, y \rangle$ is interpreted as a lower and an upper bound of the exact value of the certainty value. A similar interpretation is given elsewhere (Denecker *et al.* 1999; Denecker *et al.* 2002; Denecker *et al.* 2003), but this time $L$ is the set of two-valued interpretations, and a pair $\langle J_I^-, J_I^+ \rangle \in L \times L$ is interpreted as a lower and upper bound approximation of the application of a monotone (immediate consequence) operator $O: L \to L$ to an interpretation $I$.

Formally, given the lattice $\langle L, \preceq \rangle$, the *bilattice of intervals* is $\langle L \times L, \preceq_t, \preceq_k \rangle$, where:

- $\langle x_1, x_2 \rangle \preceq_t \langle y_1, y_2 \rangle$ if $x_1 \preceq y_1$ and $x_2 \preceq y_2$;
- $\langle x_1, x_2 \rangle \preceq_k \langle y_1, y_2 \rangle$ if $x_1 \preceq y_1$ and $y_2 \preceq x_2$.

The intuition of these orders is that truth increases if the interval contains greater values, whereas the knowledge increases when the interval becomes more precise. Negation can be defined as $\neg\langle x, y \rangle = \langle \neg y, \neg x \rangle$, where $\neg$ is a negation operator on $L$. Note that, if $L = \{\mathtt{f}, \mathtt{t}\}$, and if we assign $\mathtt{f} = \langle \mathtt{f}, \mathtt{f} \rangle$, $\mathtt{t} = \langle \mathtt{t}, \mathtt{t} \rangle$, $\bot = \langle \mathtt{f}, \mathtt{t} \rangle$ and $\top = \langle \mathtt{t}, \mathtt{f} \rangle$, then we obtain a structure that is isomorphic to the bilattice $\mathscr{FOUR}$.

### 2.3 Logic programs, interpretations, models and program knowledge completions

We recall here the definitions given in Fitting (1993). This setting is as general as possible, so that the results proved in this paper will be widely applicable.

Classical logic programming has the set $\{f, t\}$ as its truth space, but as stated by Fitting (1993), "$\mathscr{FOUR}$ can be thought as the 'home' of ordinary logic programming and its natural extension is to bilattices other than $\mathscr{FOUR}$: the more general the setting the more general the results". We will also consider bilattices as the truth space of logic programs.

### 2.3.1 Logic programs

Consider an alphabet of predicate symbols, of constants, of function symbols and variable symbols. A *term*, $t$, is either a variable $x$, a constant $c$ or of the form $f(t_1, \ldots, t_n)$, where $f$ is an $n$-ary function symbol and all $t_i$ are terms. An *atom*, $A$, is of the form $p(t_1, \ldots, t_n)$, where $p$ is an $n$-ary predicate symbol and all $t_i$ are terms. A literal, $l$, is of the form $A$ or $\neg A$, where $A$ is an atom. A *formula*, $\varphi$, is an expression built up from the literals and the members of a bilattice $\mathscr{B}$ using $\wedge, \vee, \otimes, \oplus, \exists$ and $\forall$. Note that members of the bilattice may appear in a formula, e.g. in $\mathscr{FOUR}$, $(p \wedge q) \oplus (r \otimes f)$ is a formula. A *rule* is of the form $p(x_1, \ldots, x_n) \leftarrow \varphi(x_1, \ldots, x_n)$, where $p$ is an $n$-ary predicate symbol and all $x_i$ are variables. The atom $p(x_1, \ldots, x_n)$ is called the *head*, and the formula $\varphi(x_1, \ldots, x_n)$ is called the *body*. It is assumed that the free variables of the body are among $x_1, \ldots, x_n$. Free variables are thought of as universally quantified. A *logic program*, denoted with $\mathscr{P}$, is a finite set of rules. The *Herbrand universe* of $\mathscr{P}$ is the set of *ground* (variable-free) terms that can be built from the constants and function symbols occurring in $\mathscr{P}$, while the *Herbrand base* of $\mathscr{P}$ (denoted $B_{\mathscr{P}}$) is the set of ground atoms over the Herbrand universe.

*Definition 2.8* ($\mathscr{P}^*$)
Given a logic program $\mathscr{P}$, the associated set $\mathscr{P}^*$ is constructed as follows;

1. put in $\mathscr{P}^*$ all ground instances of members of $\mathscr{P}$ (over the Herbrand base);
2. if a ground atom $A$ is not head of any rule in $\mathscr{P}^*$, then add the rule $A \leftarrow f$ to $\mathscr{P}^*$. Note that it is a standard practice in logic programming to consider such atoms as *false*. We incorporate this by explicitly adding $A \leftarrow f$ to $\mathscr{P}^*$;
3. replace several ground rules in $\mathscr{P}^*$ having same head, $A \leftarrow \varphi_1, A \leftarrow \varphi_2, \ldots$ with $A \leftarrow \varphi_1 \vee \varphi_2 \vee \ldots$. As there could be infinitely many grounded rules with same head, we may end with a countable disjunction, but the semantics behavior is unproblematic.

Note that in $\mathscr{P}^*$, each ground atom appears in the head of *exactly one* rule.

### 2.3.2 Interpretations

Let $\langle \mathscr{B}, \leq_t, \leq_k \rangle$ be a bilattice. By *interpretation of a logic program* on the bilattice we mean a mapping $I$ from ground atoms to members of $\mathscr{B}$. An interpretation $I$ is extended from atoms to formulae as follows:

1. for $b \in \mathscr{B}$, $I(b) = b$;
2. for formulae $\varphi$ and $\varphi'$, $I(\varphi \wedge \varphi') = I(\varphi) \wedge I(\varphi')$, and similarly for $\vee, \otimes, \oplus$ and $\neg$; and

3. $I(\exists x \varphi(x)) = \bigvee \{I(\varphi(t)) : t$ ground term$\}$, and similarly for universal quantification [2].

The family of all interpretations is denoted by $\mathscr{I}(\mathscr{B})$. The truth and knowledge orderings are extended from $\mathscr{B}$ to $\mathscr{I}(\mathscr{B})$ as follows:

- $I_1 \preceq_t I_2$ iff $I_1(A) \preceq_t I_2(A)$, for every ground atom $A$; and
- $I_1 \preceq_k I_2$ iff $I_1(A) \preceq_k I_2(A)$, for every ground atom $A$.

Given two interpretations $I, J$, we define $(I \wedge J)(\varphi) = I(\varphi) \wedge J(\varphi)$, and similarly for the other operations. With $\mathtt{I_f}$ and $\mathtt{I_t}$ we denote the bottom and top interpretations under $\preceq_t$ (they map any atom into $\mathtt{f}$ and $\mathtt{t}$, respectively). With $\mathtt{I_\perp}$ and $\mathtt{I_\top}$ we denote the bottom and top interpretations under $\preceq_k$ (they map any atom into $\perp$ and $\top$, respectively). It is easy to see that the space of interpretations $\langle \mathscr{I}(\mathscr{B}), \preceq_t, \preceq_k \rangle$ is an infinitary interlaced and distributive bilattice as well.

### 2.3.3 Classical setting

Note that in a *classical logic program* the body is a conjunction of literals. Therefore, if $A \leftarrow \varphi \in \mathscr{P}^*$, then $\varphi = \varphi_1 \vee \ldots \vee \varphi_n$ and $\varphi_i = L_{i_1} \wedge \ldots \wedge L_{i_n}$. Furthermore, a *classical total interpretation* is an interpretation over $\mathscr{FOUR}$ such that an atom is mapped into either $\mathtt{f}$ or $\mathtt{t}$. A *partial classical interpretation* is a classical interpretation where the truth of some atom may be left unspecified. This is the same as saying that the interpretation maps all atoms into either $\mathtt{f}, \mathtt{t}$ or $\perp$. For a set of literals $X$, with $\neg.X$ we indicate the set $\{\neg L : L \in X\}$, where for any atom $A$, $\neg\neg A$ is replaced with $A$. Then, a classical interpretation (total or partial) can also be represented as a consistent set of literals, i.e. $I \subseteq B_{\mathscr{P}} \cup \neg.B_{\mathscr{P}}$ and for all atoms $A$, $\{A, \neg A\} \nsubseteq I$. Of course, the opposite is also true, i.e. a consistent set of literals can straightforwardly be turned into an interpretation over $\mathscr{FOUR}$.

### 2.3.4 Models

An interpretation $I$ is a *model* of a logic program $\mathscr{P}$, denoted by $I \models \mathscr{P}$, if and only if for each rule $A \leftarrow \varphi$ in $\mathscr{P}^*$, $I(\varphi) \preceq_t I(A)$. With $mod(\mathscr{P})$ we identify the set of models of $\mathscr{P}$.

From all models of a logic program $\mathscr{P}$, Fitting (1993; 2002) identifies a subset, which obeys the so-called *Clark-completion* procedure (1978). Essentially, we replace in $\mathscr{P}^*$ each occurrence of $\leftarrow$ with $\leftrightarrow$: an interpretation $I$ is a *Clark-completion model*, *cl-model* for short, of a logic program $\mathscr{P}$, denoted by $I \models_{cl} \mathscr{P}$, if and only if for each rule $A \leftarrow \varphi$ in $\mathscr{P}^*$, $I(A) = I(\varphi)$. With $mod_{cl}(\mathscr{P})$ we identify the set of cl-models of $\mathscr{P}$. Of course $mod_{cl}(\mathscr{P}) \subseteq mod(\mathscr{P})$ holds.

---

[2] The bilattice is complete w.r.t. $\preceq_t$, so existential and universal quantification are well-defined.
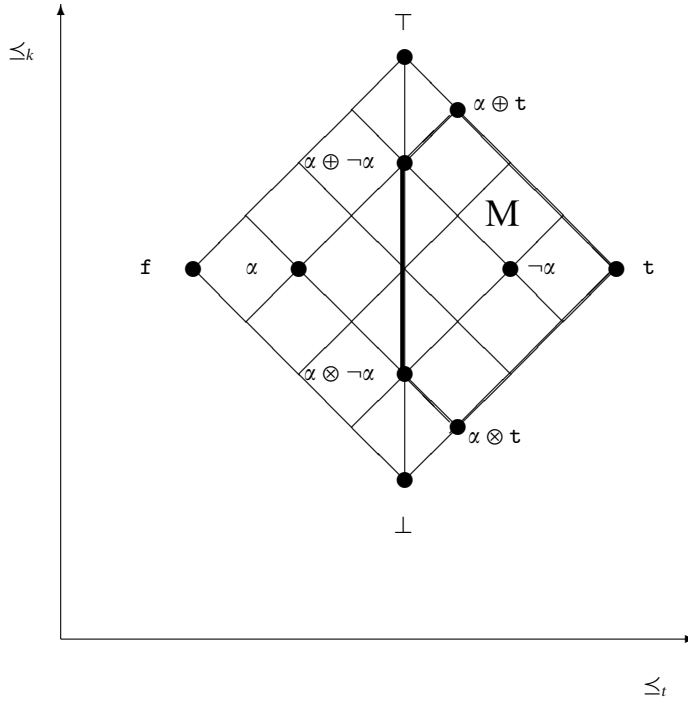
Fig. 3. Models and cl-models.

*Example 2.9*
Consider the following logic program

$$\mathscr{P} = \{(A \leftarrow \neg A), (A \leftarrow \alpha)\},$$

where $\alpha$ is a value of a bilattice such that $\alpha \preceq_t \neg \alpha$ and $A$ is a ground atom. Then $\mathscr{P}^*$ is

$$\mathscr{P}^* = \{A \leftarrow \neg A \vee \alpha\}.$$

Consider Figure 3. The set of models of $\mathscr{P}$, $mod(P)$, is the set of interpretations assigning to $A$ a value in the area ($M$-area in Figure 3) delimited by the extremal points, $\alpha \otimes \neg \alpha, \alpha \oplus \neg \alpha, \alpha \oplus t, t$ and $\alpha \otimes t$. The $\preceq_k$-least element $I$ of $mod(P)$ is such that $I(A) = \alpha \otimes t$.

The set of cl-models of $\mathscr{P}$, $mod_{cl}(P)$, is the set of interpretations assigning to $A$ a value on the vertical line, in between the extremal points $\alpha \otimes \neg \alpha$ and $\alpha \oplus \neg \alpha$ and are all truth minimal. The $\preceq_k$-least element $I'$ of $mod_{cl}(P)$ is such that $I'(A) = \alpha \otimes \neg \alpha$. Note that $I$ is not a cl-model of $\mathscr{P}$ and, thus, $mod_{cl}(P) \subset mod(P)$.

Clark-completion models also have an alternative characterisation.

*Definition 2.10* (*general reduct*)
Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. The *general reduct* of $\mathscr{P}$ w.r.t. $I$, denoted $\mathscr{P}[I]$ is the program obtained from $\mathscr{P}^*$ in which each (ground) rule $A \leftarrow \varphi \in \mathscr{P}^*$ is replaced with $A \leftarrow I(\varphi)$.

Note that any model $J$ of $\mathscr{P}[I]$ is such that for all rules $A \leftarrow \varphi \in \mathscr{P}^*$, $I(\varphi) \leq_t J(A)$. But, in $\mathscr{P}^*$ each ground atom appears in the head of exactly one rule. Therefore, it is easily verified that any $\leq_t$-minimal model $J$ of $\mathscr{P}[I]$ is such that $J(A) = I(\varphi)$ and there can be just one such model, i.e. $J = \min_{\leq_t}\{J' : J' \models \mathscr{P}[I]\}$.

We have the following theorem, which allows us to express the cl-models of a logic program in terms of its models.

*Theorem 2.11*
Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. Then $I \models_{cl} \mathscr{P}$ iff $I = \min_{\leq_t}\{J : J \models \mathscr{P}[I]\}$.

*Proof*
$I \models_{cl} \mathscr{P}$ iff for all $A \leftarrow \varphi \in \mathscr{P}^*$, $I(A) = I(\varphi)$ holds iff (as noted above) $I = \min_{\leq_t}\{J : J \models \mathscr{P}[I]\}$.   $\square$

The above theorem establishes that Clark-completion models are fixed-points of the operator $\Gamma_{\mathscr{P}} : \mathscr{I}(\mathscr{B}) \to \mathscr{I}(\mathscr{B})$, defined as

$$\Gamma_{\mathscr{P}}(I) = \min_{\leq_t}\{J : J \models \mathscr{P}[I]\} , \qquad (2)$$

i.e. $I \models_{cl} \mathscr{P}$ iff $I = \Gamma_{\mathscr{P}}(I)$.

### 2.3.5 Program knowledge completions

Finally, given an interpretation $I$, we introduce the notion of program *knowledge completion*, or simply, *k-completion* with $I$, denoted $\mathscr{P} \oplus I$. The *program k-completion* of $\mathscr{P}$ with $I$, is the program obtained by replacing any rule of the form $A \leftarrow \varphi \in \mathscr{P}$ by $A \leftarrow \varphi \oplus I(A)$. The idea is to enforce any model $J$ of $\mathscr{P} \oplus I$ to contain at least the knowledge determined by $\mathscr{P}$ and $I$. Note that $J \models \mathscr{P} \oplus I$ does not imply $J \models \mathscr{P}$. For instance, given $\mathscr{P} = \{A \leftarrow A \otimes \neg A\}$ and $I = J = \mathtt{I_f}$, then $\mathscr{P} \oplus I = \{A \leftarrow (A \otimes \neg A) \oplus \mathtt{f}\}$ and $J \models \mathscr{P} \oplus I$, while $J \not\models \mathscr{P}$.

### 2.3.6 Additional remarks

Note that the use of the negation, $\neg$, in literals has to be understood as *classical negation*. The expression *not L* (where $L$ is a literal) appearing quite often as syntactical construct in logic programs, indicating '$L$ is not provable', is not part of our language. This choice is intentional, as we want to stress that in this study CWA will be considered as an additional source of (or carrier of) falsehood in an abstract sense and will be considered as a 'cumulative' information source with the classical semantics (Kripke-Kleene semantics). In this sense, our approach is an attempt to relate the stable model semantics of logic programs to a standard model-theoretic account of rules, relying on standard logical means as the notion of interpretation and knowledge ordering.

## 2.4 Semantics of logic programs

In logic programming, usually the semantics of a program $\mathscr{P}$ is determined by selecting a particular interpretation, or a set of interpretations, of $\mathscr{P}$ in the set of models of $\mathscr{P}$. We consider three semantics, which are probably the most popular and widely studied semantics for logic programs with negation, namely *Kripke-Kleene semantics*, *well-founded semantics* and *stable model semantics*, in increasing order of knowledge.

### 2.4.1 Kripke-Kleene semantics

Kripke-Kleene semantics (Fitting 1985) has a simple, intuitive and epistemic characterization, as it corresponds to the least cl-model of a logic program under the knowledge order $\preceq_k$. Kripke-Kleene semantics is essentially a generalization of the least model characterization of classical programs without negation over the truth space $\{\mathtt{f}, \mathtt{t}\}$ (Emden and Kowalski 1976; Lloyd 1987) to logic programs with classical negation evaluated over bilattices under Clark's program completion. More formally:

*Definition 2.12* (*Kripke-Kleene semantics*)
The *Kripke-Kleene model* of a logic program $\mathscr{P}$ is the $\preceq_k$-least cl-model of $\mathscr{P}$, i.e.

$$KK(\mathscr{P}) = \min_{\preceq_k}(\{I : I \models_{cl} \mathscr{P}\}). \tag{3}$$

For instance, by referring to Example 2.9, the value of $A$ w.r.t. the Kripke-Kleene semantics of $\mathscr{P}$ is $KK(\mathscr{P})(A) = \alpha \otimes \neg\alpha$.

Note that by Theorem 2.11 and by Equation (2) we have also

$$KK(\mathscr{P}) = \mathrm{lfp}_{\preceq_k}(\Gamma_{\mathscr{P}}). \tag{4}$$

Kripke-Kleene semantics also has an alternative, and better known, fixed-point characterization, which relies on the well-known $\Phi_{\mathscr{P}}$ immediate consequence operator. $\Phi_{\mathscr{P}}$ is a generalization of the Van Emden-Kowalski's immediate consequence operator $T_{\mathscr{P}}$ (Emden and Kowalski 1976; Lloyd 1987) to bilattices under Clark's program completion. Interesting properties of $\Phi_{\mathscr{P}}$ are that (*i*) $\Phi_{\mathscr{P}}$ relies on the classical evaluation of negation, i.e. the evaluation of a negative literal $\neg A$ is given by the negation of the evaluation of $A$; and (*ii*) $\Phi_{\mathscr{P}}$ is monotone with respect to the knowledge ordering and, thus, has a $\preceq_k$-least fixed-point, which coincides with the Kripke-Kleene semantics of $\mathscr{P}$. Formally,

*Definition 2.13* (*immediate consequence operator* $\Phi_{\mathscr{P}}$)
Consider a logic program $\mathscr{P}$. The *immediate consequence operator* $\Phi_{\mathscr{P}} : \mathscr{I}(\mathscr{B}) \to \mathscr{I}(\mathscr{B})$ is defined as follows. For $I \in \mathscr{I}(\mathscr{B})$, $\Phi_{\mathscr{P}}(I)$ is the interpretation, which for any ground atom $A$ such that $A \leftarrow \varphi$ occurs in $\mathscr{P}^*$, satisfies $\Phi_{\mathscr{P}}(I)(A) = I(\varphi)$.

It can easily be shown that

*Theorem 2.14 (Fitting 1993)*

In the space of interpretations, the operator $\Phi_{\mathscr{P}}$ is monotone under $\preceq_k$, the set of fixed-points of $\Phi_{\mathscr{P}}$ is a complete lattice under $\preceq_k$ and, thus, $\Phi_{\mathscr{P}}$ has a $\preceq_k$-least fixed-point. Furthermore, $I$ is a cl-model of a program $\mathscr{P}$ iff $I$ is a fixed-point of $\Phi_{\mathscr{P}}$. Therefore, the Kripke-Kleene model of $\mathscr{P}$ coincides with $\Phi_{\mathscr{P}}$'s least fixed-point under $\preceq_k$.

For instance, by referring to Example 2.9, the set of fixed-points of $\Phi_{\mathscr{P}}$ coincides with the set of interpretations assigning to $A$ a value on the vertical line, in between the extremal points $\alpha \otimes \neg\alpha$ and $\alpha \oplus \neg\alpha$.

The above theorem relates the model theoretic and epistemic characterization of the Kripke-Kleene semantics to a least fixed-point characterization. By relying on $\Phi_{\mathscr{P}}$ we also know how to effectively compute $KK(\mathscr{P})$ as given by the Knaster-Tarski Theorem 2.1.

Note that from Theorem 2.11 and equation (2), it follows immediately that

*Corollary 2.15*

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. Then $\Phi_{\mathscr{P}}(I) = \Gamma_{\mathscr{P}}(I)$.

*Proof*

Let $I' = \Gamma_{\mathscr{P}}(I) = \min_{\preceq_t}(\{J : J \models \mathscr{P}[I]\})$. Then we have that for any ground atom $A$, $\Gamma_{\mathscr{P}}(I)(A) = I'(A) = I(\varphi) = \Phi_{\mathscr{P}}(I)(A)$, i.e. $\Phi_{\mathscr{P}}(I) = \Gamma_{\mathscr{P}}(I)$.  $\square$

As a consequence, *all definitions and properties given in this paper in terms of $\Phi_{\mathscr{P}}$ and/or cl-models may be given in terms of $\Gamma_{\mathscr{P}}$ and/or models as well.* As $\Phi_{\mathscr{P}}$ is a well-known operator, for ease of presentation we will continue use it.

We conclude this section with the following simple lemma, which will be used later in the paper.

*Lemma 2.16*

Let $\mathscr{P}$ be a logic program and let $J$ and $I$ be interpretations. Then $\Phi_{\mathscr{P}\oplus I}(J) = \Phi_{\mathscr{P}}(J) \oplus I$. In particular, $J \models_{cl} \mathscr{P} \oplus I$ iff $J = \Phi_{\mathscr{P}}(J) \oplus I$.

### 2.4.2 Stable model and well-founded semantics

A commonly accepted approach towards provide a stronger semantics or a semantics that is more informative to logic programs than the Kripke-Kleene semantics, consists in relying on CWA to complete the available knowledge. Of the various approaches to the management of negation in logic programming, the *stable model semantics* approach, introduced by Gelfond and Lifschitz (1988) with respect to the classical two valued truth space $\{\mathtt{f}, \mathtt{t}\}$ has become one of the most widely studied and most commonly accepted proposal. Informally, a set of ground atoms $I$ is a *stable model* of a classical logic program $\mathscr{P}$ if $I = I'$, where $I'$ is computed according to the so-called *Gelfond-Lifschitz transformation*:

1.  substitute (fix) in $\mathscr{P}^*$ the negative literals by their evaluation with respect to $I$. Let $\mathscr{P}^I$ be the resulting *positive* program, called *reduct* of $\mathscr{P}$ w.r.t. $I$;

2. let $I'$ be the minimal Herbrand (truth-minimal) model of $\mathcal{P}^I$.

This approach defines a whole family of models and it has been shown (Przymusinski 1990c) that the *minimal* one according to the knowledge ordering corresponds to the *well-founded semantics* (van Gelder *et al.* 1991).

The extension of the notions of stable model and well-founded semantics to the context of bilattices is due to Fitting (1993). He proposes a generalization of the Gelfond-Lifschitz transformation to bilattices by means of the binary immediate consequence operator $\Psi_{\mathcal{P}}$. Similarly to that of the Gelfond-Lifschitz transformation, the basic principle of $\Psi_{\mathcal{P}}$ is to separate the roles of positive and negative information. Informally, $\Psi_{\mathcal{P}}$ accepts two input interpretations over a bilattice, the first is used to assign meanings to positive literals, while the second is used to assign meanings to negative literals. $\Psi_{\mathcal{P}}$ is monotone in both arguments in the knowledge ordering $\preceq_k$. But, with respect to the truth ordering $\preceq_t$, $\Psi_{\mathcal{P}}$ is monotone in the first argument, while it is antitone in the second argument (indeed, as the truth of a positive literal increases, the truth of its negation decreases). Computationally, Fitting follows the idea of the Gelfond-Lifschitz transformation shown above: the idea is to fix an interpretation for negative information and to compute the $\preceq_t$-least model of the resulting positive program. To this end, Fitting 1993 additionally introduced the $\Psi'_{\mathcal{P}}$ operator, which for a given interpretation $I$ of negative literals, computes the $\preceq_t$-least model, $\Psi'_{\mathcal{P}}(I) = \mathrm{lfp}_{\preceq_t}(\lambda x.\Psi_{\mathcal{P}}(x,I))$. The fixed-points of $\Psi'_{\mathcal{P}}$ are the stable models, while the least fixed-point of $\Psi'_{\mathcal{P}}$ under $\preceq_k$ is the well-founded semantics of $\mathcal{P}$.

Formally, let $I$ and $J$ be two interpretations in the bilattice $\langle \mathscr{I}(\mathscr{B}), \preceq_t, \preceq_k \rangle$. The notion of *pseudo-interpretation* $I \triangle J$ over the bilattice is defined as follows ($I$ gives meaning to positive literals, while $J$ gives meaning to negative literals): for a pure ground atom $A$:

$$\begin{aligned} (I \triangle J)(A) &= I(A) \\ (I \triangle J)(\neg A) &= \neg J(A). \end{aligned}$$

Pseudo-interpretations are extended to non-literals in the obvious way. We can now define $\Psi_{\mathcal{P}}$ as follows.

*Definition 2.17 (immediate consequence operator $\Psi_{\mathcal{P}}$)*
The *immediate consequence operator* $\Psi_{\mathcal{P}} \colon \mathscr{I}(\mathscr{B}) \times \mathscr{I}(\mathscr{B}) \to \mathscr{I}(\mathscr{B})$ is defined as follows. For $I, J \in \mathscr{I}(\mathscr{B})$, $\Psi_{\mathcal{P}}(I,J)$ is the interpretation, which for any ground atom $A$ such that $A \leftarrow \varphi$ occurs in $\mathcal{P}^*$, satisfies $\Psi_{\mathcal{P}}(I,J)(A) = (I \triangle J)(\varphi)$.

Note that $\Phi_{\mathcal{P}}$ is a special case of $\Psi_{\mathcal{P}}$, as from construction $\Phi_{\mathcal{P}}(I) = \Psi_{\mathcal{P}}(I,I)$. The following theorem can be shown.

*Theorem 2.18 (Fitting 1993)*
In the space of interpretations the operator $\Psi_{\mathcal{P}}$ is monotone in both arguments under $\preceq_k$, and under the ordering $\preceq_t$ it is monotone in its first argument and antitone in its second argument.

We are ready now to define the $\Psi'_{\mathcal{P}}$ operator.

*Definition 2.19* (stability operator $\Psi'_{\mathscr{P}}$)
The *stability operator* of $\Psi_{\mathscr{P}}$ is the single input operator $\Psi'_{\mathscr{P}}$ given by: $\Psi'_{\mathscr{P}}(I)$ is the $\preceq_t$-least fixed-point of the operator $\lambda x.\Psi_{\mathscr{P}}(x, I)$, i.e. $\Psi'_{\mathscr{P}}(I) = \mathrm{lfp}_{\preceq_t}(\lambda x.\Psi_{\mathscr{P}}(x, I))$.

By Theorem 2.18, $\Psi'_{\mathscr{P}}$ is well defined and can be computed in the usual way: let $I$ be an interpretation. Consider the following sequence: for $i \geqslant 0$,

$$
\begin{aligned}
v_0^I &= \mathtt{I_f}, \\
v_{i+1}^I &= \Psi_{\mathscr{P}}(v_i^I, I).
\end{aligned}
$$

Then the $v_i^I$ sequence is monotone non-decreasing under $\preceq_t$ and converges to $\Psi'_{\mathscr{P}}(I)$. In the following, with $v_i^I$ we will always indicate the $i$-th iteration of the computation of $\Psi'_{\mathscr{P}}(I)$.

The following theorem holds.

*Theorem 2.20* (Fitting 1993)
The operator $\Psi'_{\mathscr{P}}$ is monotone in the $\preceq_k$ ordering, and antitone in the $\preceq_t$ ordering. Furthermore, every fixed-point of $\Psi'_{\mathscr{P}}$ is also a fixed-point of $\Phi_{\mathscr{P}}$, i.e. a cl-model of $\mathscr{P}$.

Finally, following Fitting's formulation,

*Definition 2.21* (stable model)
A *stable model* for a logic program $\mathscr{P}$ is a fixed-point of $\Psi'_{\mathscr{P}}$. With $stable(\mathscr{P})$ we indicate the set of stable models of $\mathscr{P}$.

Note that it can be seen immediately from the definition of $\Psi'_{\mathscr{P}}$ that

$$\Psi'_{\mathscr{P}}(I) = \min_{\preceq_t}(mod(P^I))\ ^3$$

and, thus,

$$I \in stable(\mathscr{P}) \text{ iff } I = \min_{\preceq_t}(mod(\mathscr{P}^I)). \tag{5}$$

By Theorem 2.20 and the Knaster-Tarski Theorem 2.1, the set of fixed-points of $\Psi'_{\mathscr{P}}$, i.e. the set of stable models of $\mathscr{P}$, is a complete lattice under $\preceq_k$ and, thus, $\Psi'_{\mathscr{P}}$ has a $\preceq_k$-least fixed-point, which is denoted $WF(\mathscr{P})$. $WF(\mathscr{P})$ is known as the *well-founded model* of $\mathscr{P}$ and, by definition, coincides with the $\preceq_k$-least stable model, i.e.

$$WF(\mathscr{P}) = \min_{\preceq_k}(\{I : I \text{ stable model of } \mathscr{P}\}). \tag{6}$$

The characterization of the well-founded model in terms of least fixed-point of $\Psi'_{\mathscr{P}}$ also gives us a way to effectively compute it.

It is interesting to note, that for classical logic programs the original definition of well-founded semantics is based on the well-known notion of *unfounded set* (van Gelder *et al.* 1991). The underlying principle of the notion of unfounded sets is to identify the set of atoms that can safely be assumed false if the current information about a logic program is given by an interpretation $I$. Indeed, given a classical interpretation $I$ and a classical logic program $\mathscr{P}$, a set of ground atoms $X \subseteq B_{\mathscr{P}}$ is

---

³ As $\mathscr{P}^I$ is positive, it has a unique truth-minimal model.

an *unfounded set* (i.e., the atoms in $X$ can be assumed as false) for $\mathscr{P}$ w.r.t. $I$ iff for each atom $A \in X$,

1. if $A \leftarrow \varphi \in \mathscr{P}^*$ (note that $\varphi = \varphi_1 \vee \ldots \vee \varphi_n$ and $\varphi_i = L_{i_1} \wedge \ldots \wedge L_{i_n}$), then $\varphi_i$ is false either w.r.t. $I$ or w.r.t. $\neg.X$, for all $1 \leqslant i \leqslant n$.

A well-known property of unfounded sets is that the union of two unfounded sets of $\mathscr{P}$ w.r.t. $I$ is an unfounded set as well and, thus, there is a unique *greatest unfounded set* for $\mathscr{P}$ w.r.t. $I$, denoted by $U_{\mathscr{P}}(I)$.

Now, consider the usual immediate consequence operator $T_{\mathscr{P}}$, where for any ground atom $A$,

$$T_{\mathscr{P}}(I)(A) = \mathtt{t} \text{ iff } \exists A \leftarrow \varphi \in \mathscr{P}^* \text{ s.t. } I(\varphi) = \mathtt{t},$$

and consider the well-founded operator (van Gelder *et al.*, 1991) over classical interpretations $I$

$$W_{\mathscr{P}}(I) = T_{\mathscr{P}}(I) \cup \neg.U_{\mathscr{P}}(I). \tag{7}$$

$W_{\mathscr{P}}(I)$ can be rewritten as $W_{\mathscr{P}}(I) = T_{\mathscr{P}}(I) \oplus \neg.U_{\mathscr{P}}(I)$, by assuming $\oplus = \cup, \otimes = \cap$ in the lattice $\langle 2^{B_{\mathscr{P}} \cup \neg.B_{\mathscr{P}}}, \subseteq \rangle$ (the partial order $\subseteq$ corresponds to the knowledge order $\preceq_k$). Then,

- the well-founded semantics is defined to be the $\preceq_k$-least fixed-point of $W_{\mathscr{P}}$ in van Gelder *et al.* (1991), and
- it is shown in Leone (1997) that the set of *total* stable models of $\mathscr{P}$ coincides with the set of total fixed-points of $W_{\mathscr{P}}$.

In particular, this formulation reveals that the greatest unfounded set, $\neg.U_{\mathscr{P}}(I)$, is the additional "false default knowledge", which is introduced by CWA into the usual semantics of logic programs given by $T_{\mathscr{P}}$. However, $W_{\mathscr{P}}$ does not allow partial stable models to be identified. Indeed, there are fixed-points of $W_{\mathscr{P}}(I)$ that are partial interpretations, which are not stable models.

We conclude the preliminary part of the paper with the following result that adds to Fitting's analysis that stable models are incomparable with each other with respect to the truth order $\preceq_t$.

*Theorem 2.22*
Let $I$ and $J$ be two stable models such that $I \neq J$. Then $I \npreceq_t J$ and $J \npreceq_t I$.

*Proof*
Assume to the contrary that either $I \preceq_t J$ or $J \preceq_t I$ holds. Without loss of generality, assume $I \preceq_t J$. By Theorem 2.20, $\Psi'_{\mathscr{P}}$ is antitone in the $\preceq_t$ ordering. Therefore, from $I \preceq_t J$ it follows that $J = \Psi'_{\mathscr{P}}(J) \preceq_t \Psi'_{\mathscr{P}}(I) = I$ holds and, thus, $I = J$, a contradiction to the hypothesis. $\square$

## 3 Stable model semantics revisited

In the following, by relying on CWA as a source of falsehood for knowledge completion, we provide epistemic and fixed-point based, characterizations of the

well-founded and stable model semantics over bilattices that are alternative to the one provided by Fitting (1993). We proceed in three steps.

(*i*) In the next section, we introduce the notion of *support*, denoted $s_{\mathscr{P}}(I)$, with respect to a logic program $\mathscr{P}$ and an interpretation $I$. The support is a generalization of the notion of greatest unfounded set (which determines the atoms that can be assumed to be false) w.r.t. $I$ from classical logic programming to bilattices. Intuitively, we regard CWA as an additional source of information for *falsehood* to be used to complete $I$. The support $s_{\mathscr{P}}(I)$ of $\mathscr{P}$ w.r.t. $I$ determines in a principled way the amount, or *degree*, of falsehood provided by CWA to the atom's truth that can be added to current knowledge $I$ about the program $\mathscr{P}$. It turns out that for classical logic programs the support coincides with the negation of the greatest unfounded set, i.e. $s_{\mathscr{P}}(I) = \neg . U_{\mathscr{P}}(I)$.

(*ii*) Any model $I$ of $\mathscr{P}$ containing its support, i.e. such that $s_{\mathscr{P}}(I) \leq_k I$, tells us that the additional source of falsehood provided by CWA cannot contribute improving our knowledge about the program $\mathscr{P}$. We call such models *supported models* of $\mathscr{P}$; this will be discussed in Section 3.2. Supported models can be characterized as fixed-points of the operator

$$\tilde{\Pi}_{\mathscr{P}}(I) = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I) \,,$$

which is very similar to the $W_{\mathscr{P}}$ operator in Equation (7), but generalized to bilattices. As expected, it can be shown that the $\leq_k$-*least supported model is the well-founded model of* $\mathscr{P}$. Unfortunately, while for classical logic programs and total interpretations, supported models characterize total stable models (in fact, they coincides with the fixed-points of $W_{\mathscr{P}}$), this is not true in the general case of interpretations over bilattices.

Therefore, we further refine the class of supported models, by introducing the class of *models deductively closed under support k-completion*. This class requires supported models to satisfy some minimality condition with respect to the knowledge order $\leq_k$. Indeed, such a model $I$ has to be deductively closed according to the Kripke-Kleene semantics of the program $k$-completed with its support, i.e.

$$I = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I)) \tag{8}$$

or, equivalently

$$I = \min_{\leq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I))). \tag{9}$$

(*iii*) We will show that any such interpretation $I$ is a stable model of $\mathscr{P}$ and vice-versa, i.e. $I \in stable(\mathscr{P})$ iff $I = \min_{\leq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I)))$, which is quite suggestive. Note that until now, stable models (over bilattices) have been characterized as by equation (5). Equation (9) above shows thus that stable models can be characterized as those models that contain their support and are deductively closed under the Kripke-Kleene semantics. As such, we can identify the support (unfounded set, in classical terms) as the added-value (in terms of knowledge), which is brought into by the stable model semantics with respect to the standard Kripke-Kleene semantics of $\mathscr{P}$.

Table 1. *Models, Kripke-Kleene, well-founded and stable models of* $\mathscr{P}$

| $I_i \models_{cl} \mathscr{P}$ | $p$ | $q$ | $r$ | $KK(\mathscr{P})$ | $WF(\mathscr{P})$ | stable models |
|---|---|---|---|---|---|---|
| $I_1$ | $\bot$ | $\bot$ | $\bot$ | • | | |
| $I_2$ | $\bot$ | t | f | | | |
| $I_3$ | f | $\bot$ | $\bot$ | | • | • |
| $I_4$ | f | f | t | | | • |
| $I_5$ | f | t | f | | | • |
| $I_6$ | f | $\top$ | $\top$ | | | • |
| $I_7$ | t | t | f | | | |
| $I_8$ | $\top$ | t | f | | | |
| $I_9$ | $\top$ | $\top$ | $\top$ | | | |

Finally, stable models can thus be defined in terms of fixed-points of the operator $KK(\mathscr{P} \oplus s_{\mathscr{P}}(\cdot))$, which relies on a, though intuitive, program transformation $\mathscr{P} \oplus s_{\mathscr{P}}(\cdot)$. We further introduce a new operator $\Phi'_{\mathscr{P}}$, which we show to have the property that $\Phi'_{\mathscr{P}}(I) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I))$. This operator clearly shows that a model is a stable model iff it contains exactly the knowledge obtained by activating the rules over its support, without any other extra knowledge. An important property of $\Phi'_{\mathscr{P}}$ is that it does depend on $\Phi_{\mathscr{P}}$ only. This may be important in the classical logic programming case where $\mathscr{P} \oplus s_{\mathscr{P}}(\cdot)$ is not easy to define (as $\oplus$ does not belong to the language of classical logic programs). As a consequence, no program transformation is required, which completes our analysis.

We will rely on the following running example to illustrate the concepts that will be introduced in the next sections.

*Example 3.1* (*running example*)
Consider the following logic program $\mathscr{P}$ with the following rules.

$$p \leftarrow p$$
$$q \leftarrow \neg r$$
$$r \leftarrow \neg q \wedge \neg p$$

In Table 1 we report the cl-models $I_i$, the Kripke-Kleene, the well-founded and the stable models of $\mathscr{P}$, marked by bullets. Note that according to Theorem 2.22, stable models cannot be compared with each other under $\preceq_t$, while under the knowledge order, $I_3$ is the least informative model (i.e. the well-founded model), while $I_6$ is the most informative one ($I_4$ and $I_5$ are incomparable under $\preceq_k$).

### 3.1 Support

The main notion we introduce here is that of *support* of a logic program $\mathscr{P}$ with respect to a given interpretation $I$. If $I$ represents what we already know about an intended model of $\mathscr{P}$, the support represents the $\preceq_k$-greatest amount/degree of falsehood provided by CWA that can be joined to $I$ in order to complete $I$.

Falsehood is always represented in terms of an interpretation, which we call a *safe* interpretation. The main principle underlying safe interpretations can be explained as follows. For ease, let us consider $\mathscr{FOUR}$. Consider an interpretation $I$, which is our current knowledge about $\mathscr{P}$. Let us assume that the interpretation $J$, with $J \preceq_k I_{\mathtt{f}}$, indicates which atoms may be assumed as $\mathtt{f}$. For any ground atom $A$, $J(A)$ is the default 'false' information provided by $J$ to the atom $A$. The completion of $I$ with $J$ is the interpretation $I \oplus J$. In order to accept this completion, we have to ensure that the assumed false knowledge about $A$, $J(A)$, is entailed by $\mathscr{P}$ w.r.t. the completed interpretation $I \oplus J$, i.e. for $A \leftarrow \varphi \in \mathscr{P}^*$, $J(A) \preceq_k (I \oplus J)(\varphi)$ should hold. That is, after completing the current knowledge $I$ about $\mathscr{P}$ with the 'falsehood' assumption $J$, the inferred information about $A$, $(I \oplus J)(\varphi)$, should increase. Formally:

*Definition 3.2* (*safe interpretation*)

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. An interpretation $J$ is *safe* w.r.t. $\mathscr{P}$ and $I$ iff:

1. $J \preceq_k I_{\mathtt{f}}$;
2. $J \preceq_k \Phi_{\mathscr{P}}(I \oplus J)$.

As anticipated, safe interpretations have an interesting reading once we restrict our attention to the classical framework of logic programming: indeed, the concept of safe interpretation reduces to that of unfounded set.

*Theorem 3.3*

Let $\mathscr{P}$ and $I$ be a classical logic program and a classical interpretation, respectively. Let $X$ be a subset of $B_{\mathscr{P}}$. Then $X$ is an unfounded set of $\mathscr{P}$ w.r.t. $I$ iff $\neg.X \preceq_k \Phi_{\mathscr{P}}(I \oplus \neg.X)$ [4], i.e. $\neg.X$ is safe w.r.t. $\mathscr{P}$ and $I$.

*Proof*

Assume $\neg A \in \neg.X$ ( i.e. $\neg.X(\neg A) = \mathtt{t}$) and, thus, $A \in X$ (i.e. $X(A) = \mathtt{f}$). Therefore, by definition of unfounded sets, if $A \leftarrow \varphi \in \mathscr{P}^*$, where $\varphi = \varphi_1 \vee \ldots \vee \varphi_n$ and $\varphi_i = L_{i_1} \wedge \ldots \wedge L_{i_n}$, then either $I(\varphi_i) = \mathtt{f}$ or $\neg.X(\varphi_i) = \mathtt{f}$. Therefore, $(I \cup \neg.X)(\varphi) = \mathtt{f}$, i.e. $(I \oplus \neg.X)(\varphi) = \mathtt{f}$. But then, by definition of $\Phi_{\mathscr{P}}$, we have that $\Phi_{\mathscr{P}}(I \oplus \neg.X)(A) = \mathtt{f}$, i.e. $\Phi_{\mathscr{P}}(I \oplus \neg.X)(\neg A) = \mathtt{t}$. Therefore, $\neg.X \preceq_k \Phi_{\mathscr{P}}(I \oplus \neg.X)$. The other direction can be shown similarly. $\square$

The following example illustrates the concept.

*Example 3.4* (*running example cont.*)

Let us consider $I_2$. $I_2$ dictates that $p$ is unknown, $q$ is true and that $r$ is false.

---

[4] Note that this condition can be rewritten as $\neg.X \subseteq \Phi_{\mathscr{P}}(I \cup \neg.X)$.

Consider the interpretations $J_i$ defined as follows:

| $J_i$ | $p$ | $q$ | $r$ |
|-------|-----|-----|-----|
| $J_1$ | $\perp$ | $\perp$ | $\perp$ |
| $J_2$ | f | $\perp$ | $\perp$ |
| $J_3$ | $\perp$ | $\perp$ | f |
| $J_4$ | f | $\perp$ | f |

It is easy to verify that $J_i \preceq_k I_{\mathbf{f}}$ and $J_i \preceq_k \Phi_{\mathscr{P}}(I_2 \oplus J_i)$. Therefore, all the $J_i$s are safe. The $\preceq_k$-least safe interpretation is $J_1$, while the $\preceq_k$-greatest safe interpretation is $J_4 = J_1 \oplus J_2 \oplus J_3$. $J_4$ dictates that under $I_2$, we can 'safely' assume that both $p$ and $r$ are false. Note that if we join $J_4$ to $I_2$ we obtain the stable model $I_5$, where $I_2 \preceq_k I_5$. Thus, $J_4$ improves the knowledge expressed by $I_2$.

It might be asked why we do not consider $q$ false as well. In fact, if we consider $p, q$ and $r$ false, after joining to $I$ and applying $\Phi_{\mathscr{P}}$, $q$ becomes true, which is *knowledge-incompatible* with $q$'s previous knowledge status ($q$ is false). So, $q$'s falsehood is not preserved.

We also consider another example on a more general bilattice allowing the management of uncertainty.

*Example 3.5*
Let us consider the lattice $\langle L, \leq \rangle$, where $L$ is the unit interval $[0,1]$ and $\leq$ is the natural linear order $\leqslant$. The negation operator on $L$ considered is defined as $\neg x = 1 - x$. We further build the bilattice of intervals $\langle [0,1] \times [0,1], \preceq_t, \preceq_k \rangle$ in the standard way. An interval $\langle x, y \rangle$ may be understood as an approximation of the certainty of an atom.

Let us note that for $x, x', y, y' \in L$,

- $\langle x, y \rangle \wedge \langle x', y' \rangle = \langle \min(x, x'), \min(y, y') \rangle$;
- $\langle x, y \rangle \vee \langle x', y' \rangle = \langle \max(x, x'), \max(y, y') \rangle$;
- $\langle x, y \rangle \otimes \langle x', y' \rangle = \langle \min(x, x'), \max(y, y') \rangle$;
- $\langle x, y \rangle \oplus \langle x', y' \rangle = \langle \max(x, x'), \min(y, y') \rangle$; and
- $\neg \langle x, y \rangle = \langle 1 - y, 1 - x \rangle$.

Consider the logic program $\mathscr{P}$ with rules

$$A \leftarrow A \wedge C$$
$$B \leftarrow B \vee \neg C$$
$$C \leftarrow C \vee D$$
$$D \leftarrow [0.7, 0.7]$$

The fourth rule asserts that the truth value of $D$ is exactly 0.7. Then using the third rule, we will infer that the value of $C$ is given by the disjunction of 0.7 and the value of $C$ itself which is initially unknown, i.e. between 0 and 1, thus our knowledge about

$C$ is that its value is at least 0.7, i.e. [0.7; 1]. Activating the second rule with that knowledge, then the value of $B$ is given by the disjunction of the value of $\neg C$, that is at most 0.3, i.e. [0; 0.3], and the value of $B$ itself that is unknown, thus $B$ remains unknown. Similarly, the first rule does not provide any knowledge about the value of $A$. That knowledge corresponds to the Kripke-Kleene model $I$ of $\mathscr{P}$, obtained by iterating $\Phi_{\mathscr{P}}$ starting with $\mathtt{I}_{\perp} : I(A) = [0; 1], I(B) = [0; 1], I(C) = [0.7; 1]$ and $I(D) = [0.7; 0.7]$.

Relying on CWA, we should be able to provide a more precise characterization of $A$, $B$ and $C$. It can be verified that it may be safely assumed that $A$ is false ([0; 0]) and that the truth of $B$ and $C$ is at most 0.3 and 0.7, respectively, which combined with $I$ determines a more precise interpretation where $A$ is false, $B$ is at most 0.3, $C$ is 0.7 and $D$ is 0.7, respectively, as highlighted in the following table. Consider interpretations $I, J_1, J_2, J'$.

|            | $A$    | $B$        | $C$        | $D$        |
|------------|--------|------------|------------|------------|
| $I$        | [0; 1] | [0; 1]     | [0.7; 1]   | [0.7; 0.7] |
| $J_1$      | [0; 0] | [0; 1]     | [0; 0.8]   | [0; 0.7]   |
| $J_2$      | [0; 1] | [0; 0.3]   | [0; 0.7]   | [0; 1]     |
| $J'$       | [0; 0] | [0; 0.3]   | [0; 0.7]   | [0; 0.7]   |
| $I \oplus J'$ | [0; 0] | [0.0; 0.3] | [0.7; 0.7] | [0.7; 0.7] |

Both $J_1$ and $J_2$ are safe w.r.t. $\mathscr{P}$ and $I$. It is easy to see that $J' = J_1 \oplus J_2$ is the $\preceq_k$-greatest safe interpretation. Interestingly, note how $J'$ provides to $I$ some additional information on the values of $A$, $B$ and $C$, respectively.

Of all possible safe interpretations w.r.t. $\mathscr{P}$ and $I$, we are interested in the maximal one under $\preceq_k$, which is unique. The $\preceq_k$-greatest safe interpretation will be called the support provided by CWA to $\mathscr{P}$ w.r.t. $I$.

*Definition 3.6 (support)*

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. The *support provided by CWA to $\mathscr{P}$ w.r.t. $I$, or simply support of $\mathscr{P}$ w.r.t. $I$*, denoted $s_{\mathscr{P}}(I)$, is the $\preceq_k$-greatest safe interpretation w.r.t. $\mathscr{P}$ and $I$, and is given by

$$s_{\mathscr{P}}(I) = \bigoplus \{J : J \text{ is safe w.r.t. } \mathscr{P} \text{ and } I\}.$$

It is easy to show that support is a well-defined concept. Consider $X = \{J : J$ is safe w.r.t. $\mathscr{P}$ and $I\}$. As the bilattice is a complete lattice under $\preceq_k$, $lub_{\preceq_k}(X) = \oplus_{J \in X} J$ and, thus, by definition $s_{\mathscr{P}}(I) = lub_{\preceq_k}(X)$. Now consider $J \in X$. Therefore $J \preceq_k s_{\mathscr{P}}(I)$. But $J$ is safe, so $J \preceq_k \mathtt{I_f}$ and $J \preceq_k \Phi_{\mathscr{P}}(I \oplus J) \preceq_k \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I))$ (by $\preceq_k$-monotonicity of $\Phi_{\mathscr{P}}$). As a consequence, both $\mathtt{I_f}$ and $\Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I))$ are upper bounds of $X$. But $s_{\mathscr{P}}(I)$ is the least upper bound of $X$ and, thus, $s_{\mathscr{P}}(I) \preceq_k \mathtt{I_f}$ and $s_{\mathscr{P}}(I) \preceq_k \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I))$ follows. That is, $s_{\mathscr{P}}(I)$ is safe and the $\preceq_k$-greatest safe interpretation w.r.t. $\mathscr{P}$ and $I$.

It follows immediately from Theorem 3.3 that, in the classical setting, the notion of greatest unfounded set is captured by the notion of support, i.e. the support tells

Table 2. *Running example cont.: support of $\mathcal{P}$ w.r.t. $I_i$*

| $I_i \models_{cl} \mathcal{P}$ | $I_i$ | | | $s_{\mathcal{P}}(I_i)$ | | | $U_{\mathcal{P}}(I_i)$ | $KK(\mathcal{P})$ | $WF(\mathcal{P})$ | stable models |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | $p$ | $q$ | $r$ | $p$ | $q$ | $r$ | | | | |
| $I_1$ | $\bot$ | $\bot$ | $\bot$ | f | $\bot$ | $\bot$ | $\{p\}$ | • | | |
| $I_2$ | $\bot$ | t | f | f | $\bot$ | f | $\{p,r\}$ | | | |
| $I_3$ | f | $\bot$ | $\bot$ | f | $\bot$ | $\bot$ | $\{p\}$ | | • | • |
| $I_4$ | f | f | t | f | f | $\bot$ | $\{p,q\}$ | | | • |
| $I_5$ | f | t | f | f | $\bot$ | f | $\{p,r\}$ | | | • |
| $I_6$ | f | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | • |
| $I_7$ | t | t | f | f | $\bot$ | f | $\{p,r\}$ | | | |
| $I_8$ | $\top$ | t | f | f | $\bot$ | f | $\{p,r\}$ | | | |
| $I_9$ | $\top$ | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | |

us which atoms may be safely assumed to be false, given a classical interpretation $I$ and a classical logic program $\mathcal{P}$. Therefore, the notion of support extends the notion of greatest unfounded sets from the classical setting to bilattices.

*Corollary 3.7*
Let $\mathcal{P}$ and $I$ be a classical logic program and a classical interpretation, respectively. Then $s_{\mathcal{P}}(I) = \neg.U_{\mathcal{P}}(I)$.

*Example 3.8* (*running example cont.*)
Table 2 extends Table 1 also by including the supports $s_{\mathcal{P}}(I_i)$. Note that, according to Corollary 3.7, $s_{\mathcal{P}}(I_i) = \neg.U_{\mathcal{P}}(I_i)$.

Having defined the support model-theoretically, we next show how the support can effectively be computed as the iterated fixed-point of a function, $\sigma_{\mathcal{P}}^{I}$, that depends on $\Phi_{\mathcal{P}}$ only. Intuitively, the iterated computation weakens $\mathbf{I_f}$, i.e. CWA, until we arrive to the $\preceq_k$-greatest safe interpretation, i.e. the support.

*Definition 3.9* (*support function*)
Let $\mathcal{P}$ and $I$ be a logic program and an interpretation, respectively. The *support function*, denoted $\sigma_{\mathcal{P}}^{I}$, w.r.t. $P$ and $I$ is the function mapping interpretations into interpretations defined as follows: for any interpretation $J$,

$$\sigma_{\mathcal{P}}^{I}(J) \;=\; \mathbf{I_f} \otimes \Phi_{\mathcal{P}}(I \oplus J).$$

It is easy to verify that $\sigma_{\mathcal{P}}^{I}$ is monotone w.r.t. $\preceq_k$. The following theorem determines how to compute the support.

*Theorem 3.10*
Let $\mathcal{P}$ and $I$ be a logic program and an interpretation, respectively. Consider the iterated sequence of interpretations $F_i^{I}$ defined as follows: for any $i \geqslant 0$,

$$F_0^{I} \;=\; \mathbf{I_f},$$
$$F_{i+1}^{I} \;=\; \sigma_{\mathcal{P}}^{I}(F_i^{I}).$$

The sequence $F_i^I$ is

1. monotone non-increasing under $\preceq_k$ and, thus, reaches a fixed-point $F_\lambda^I$, for a limit ordinal $\lambda$; and

2. is monotone non-decreasing under $\preceq_t$.

Furthermore, $s_{\mathscr{P}}(I) = F_\lambda^I$ holds.

*Proof*

The proof is by induction. Concerning Point 1., $F_1^I \preceq_k F_0^I$; for all successor ordinal i, as $\sigma_{\mathscr{P}}^I$ is monotone under $\preceq_k$, if $F_{i+1}^I \preceq_k F_i^I$ then $F_{i+2}^I \preceq_k F_{i+1}^I$; and for all limit ordinal $\lambda$, if $i < \lambda$ then $F_\lambda^I = \bigotimes_{i < \lambda} F_i^I \preceq_k F_i^I$. Thus the sequence is monotone non-increasing under $\preceq_k$. Therefore, the sequence has a fixed-point at the limit, say $F_\lambda^I$.

Concerning Point 2., $F_0^I \preceq_t F_1^I$; for all successor ordinal i, from $F_{i+1}^I \preceq_k F_i^I$, by Lemma 2.4, we have $F_i^I = F_i^I \otimes \mathtt{I_f} \preceq_t F_{i+1}^I$; and similarly, for all limit ordinal $\lambda$, if $i < \lambda$, we have $F_\lambda^I \preceq_k F_i^I$, and by Lemma 2.4, $F_i^I = F_i^I \otimes \mathtt{I_f} \preceq_t F_\lambda^I$.

Let us show that $F_\lambda^I$ is safe and $\preceq_k$-greatest. $F_\lambda^I = \sigma_{\mathscr{P}}^I(F_\lambda^I) = \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus F_\lambda^I)$. Therefore, $F_\lambda^I \preceq_k \mathtt{I_f}$ and $F_\lambda^I \preceq_k \Phi_{\mathscr{P}}(I \oplus F_\lambda^I)$, so $F_\lambda^I$ is safe w.r.t. $\mathscr{P}$ and $I$.

Consider any $X$ safe w.r.t. $\mathscr{P}$ and $I$. We show by induction on $i$ that $X \preceq_k F_i^I$ and, thus, at the limit $X \preceq_k F_\lambda^I$, so $F_\lambda^I$ is $\preceq_k$-greatest.

(*i*) Case $i = 0$. By definition, $X \preceq_k \mathtt{I_f} = F_0^I$.

(*ii*) Induction step: suppose $X \preceq_k F_i^I$. Since $X$ is safe, we have $X \preceq_k X \otimes X \preceq_k \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus X)$. By induction, using the monotonicity of $\sigma_{\mathscr{P}}^I$ w.r.t. $\preceq_k$, $X \preceq_k \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus F_i^I) = F_{i+1}^I$.

(*iii*) Transfinite induction: given an ordinal limit $\lambda$, suppose $X \preceq_k F_i^I$ holds for all $i < \lambda$. Using the fact that the space of interpretations $\langle \mathscr{I}(\mathscr{B}), \preceq_t, \preceq_k \rangle$ is an infinitary interlaced bilattice, we have $X \preceq_k \bigotimes_{i < \lambda} F_i^I = F_\lambda^I$, which concludes the proof.     $\square$

In the following, with $F_i^I$ we indicate the *i*th iteration of the computation of the support of $\mathscr{P}$ w.r.t. $I$, according to Theorem 3.10.

Note that by construction

$$s_{\mathscr{P}}(I) = \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)), \tag{10}$$

which establishes also that the support is deductively closed in terms of falsehood. In fact, even if we add all that we know about the atom's falsehood to the current interpretation $I$, we know no more about the atom's falsehood than we knew before.

Interestingly, for a classical logic program $\mathscr{P}$ and a classical interpretation $I$, by Corollary 3.7, the above method gives us a simple top-down method to compute the negation of the greatest unfounded set, $\neg.U_{\mathscr{P}}(I)$, as the limit of the sequence:

$$\begin{aligned} F_0^I &= \neg.B_{\mathscr{P}}, \\ F_{i+1}^I &= \neg.B_{\mathscr{P}} \cap \Phi_{\mathscr{P}}(I \cup F_i^I). \end{aligned}$$

The support $s_{\mathscr{P}}(I)$ can be seen as an operator over the space of interpretations. The following theorem asserts that the support is monotone w.r.t. $\preceq_k$.

*Theorem 3.11*
Let $\mathscr{P}$ be a logic program. The support operator $s_{\mathscr{P}}$ is monotone w.r.t. $\preceq_k$.

*Proof*
Consider two interpretations $I$ and $J$, where $I \preceq_k J$. Consider the two sequences $F_i^I$ and $F_i^J$. We show by induction on $i$ that $F_i^I \preceq_k F_i^J$ and, thus, at the limit $s_{\mathscr{P}}(I) \preceq_k s_{\mathscr{P}}(J)$.

(*i*) Case $i = 0$. By definition, $F_0^I = \mathtt{I_f} \preceq_k \mathtt{I_f} = F_0^J$.

(*ii*) Induction step: suppose $F_i^I \preceq_k F_i^J$. By monotonicity under $\preceq_k$ of $\Phi_{\mathscr{P}}$ and the induction hypothesis, $F_{i+1}^I = \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus F_i^I) \preceq_k \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(J \oplus F_i^J) = F_{i+1}^J$, which concludes. $\square$

The next corollary follows directly from Lemma 2.4.

*Corollary 3.12*
Let $\mathscr{P}$ be a logic program and consider two interpretations $I$ and $J$ such that $I \preceq_k J$. Then $s_{\mathscr{P}}(J) \preceq_t s_{\mathscr{P}}(I)$.

### 3.2 Models based on the support

Of all possible models of a program $\mathscr{P}$, we are especially interested in those models $I$ that already integrate their own support, i.e. that could not be completed by CWA.

*Definition 3.13* (*supported model*)
Consider a logic program $\mathscr{P}$. An interpretation $I$ is a *supported model* of $\mathscr{P}$ iff $I \models_{cl} \mathscr{P}$ and $s_{\mathscr{P}}(I) \preceq_k I$.

If we consider the definition of support in the classical setting, then supported models are classical models of classical logic programs such that $\neg.U_{\mathscr{P}}(I) \subseteq I$, i.e. the false atoms provided by the greatest unfounded set are already false in the interpretation $I$. Therefore, CWA does not further contribute improving $I$'s knowledge about the program $\mathscr{P}$.

*Example 3.14* (*running example cont.*)
Table 3 extends Table 2 by also including supported models. Note that while both $I_8$ and $I_9$ are models of $\mathscr{P}$ including their support, they are not stable models. Note also that $s_{\mathscr{P}}(I_8) = s_{\mathscr{P}}(I_5)$ and $s_{\mathscr{P}}(I_9) = s_{\mathscr{P}}(I_6)$. That is, $I_8$ and $I_9$, which are not stable models, have the same support of some stable model.

Supported models have interesting properties, as stated below.

*Theorem 3.15*
Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. The following statements are equivalent:

1. $I$ is a supported model of $\mathscr{P}$;
2. $I = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I)$;
3. $I \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$;
4. $I = \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I))$.

Table 3. *Running example cont.: supported models of $\mathscr{P}$*

| $I_i \models_{cl} \mathscr{P}$ | $I_i$ | | | $s_{\mathscr{P}}(I_i)$ | | | $U_{\mathscr{P}}(I_i)$ | $KK(\mathscr{P})$ | $WF(\mathscr{P})$ | stable models | supported models |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $p$ | $q$ | $r$ | $p$ | $q$ | $r$ | | | | | |
| $I_1$ | $\perp$ | $\perp$ | $\perp$ | f | $\perp$ | $\perp$ | $\{p\}$ | • | | | |
| $I_2$ | $\perp$ | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | |
| $I_3$ | f | $\perp$ | $\perp$ | f | $\perp$ | $\perp$ | $\{p\}$ | | • | • | • |
| $I_4$ | f | f | t | f | f | $\perp$ | $\{p,q\}$ | | | • | • |
| $I_5$ | f | t | f | f | $\perp$ | f | $\{p,r\}$ | | | • | • |
| $I_6$ | f | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | • | • |
| $I_7$ | t | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | • |
| $I_8$ | $\top$ | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | • |
| $I_9$ | $\top$ | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | | • |

*Proof*

Assume Point 1. holds, i.e. $I \models_{cl} \mathscr{P}$ and $s_{\mathscr{P}}(I) \preceq_k I$. Then, $I = \Phi_{\mathscr{P}}(I) = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I)$, so Point 2. holds.

Assume Point 2. holds. Then, by Lemma 2.16, $I = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I) = \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(I)$, i.e. $I \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$, so Point 3. holds.

Assume Point 3. holds. So, $s_{\mathscr{P}}(I) \preceq_k I$ and from the safeness of $s_{\mathscr{P}}(I)$, it follows that $s_{\mathscr{P}}(I) \preceq_k \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)) = \Phi_{\mathscr{P}}(I)$ and, thus, $I = \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(I) = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I) = \Phi_{\mathscr{P}}(I)$. Therefore, $\Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)) = \Phi_{\mathscr{P}}(I) = I$, so Point 4. holds.

Finally, assume Point 4. holds. From the safeness of $s_{\mathscr{P}}(I)$, it follows that $s_{\mathscr{P}}(I) \preceq_k \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)) = I$. Therefore, $I = \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)) = \Phi_{\mathscr{P}}(I)$ and, thus $I$ is a supported model of $\mathscr{P}$. So, Point 1. holds, which concludes the proof. $\square$

The above theorem states the same concept in different ways: supported models contain the amount of knowledge expressed by the program and their support.

From a fixed-point characterization point of view, from Theorem 3.15 it follows that the set of supported models can be identified by the fixed-points of the $\preceq_k$-monotone operators $\Pi_{\mathscr{P}}$ and $\tilde{\Pi}_{\mathscr{P}}$ defined by

$$\Pi_{\mathscr{P}}(I) \;=\; \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)), \tag{11}$$

$$\tilde{\Pi}_{\mathscr{P}}(I) \;=\; \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I). \tag{12}$$

It follows immediately that

*Theorem 3.16*

Let $\mathscr{P}$ be a logic program. Then $\tilde{\Pi}_{\mathscr{P}}$ ($\Pi_{\mathscr{P}}$) is monotone under $\preceq_k$. Furthermore, an interpretation $I$ is a supported model iff $I = \tilde{\Pi}_{\mathscr{P}}(I)$ ($I = \Pi_{\mathscr{P}}(I)$) and, thus, relying on the Knaster-Tarski fixed-point theorem (Theorem 2.1), the set of supported models is a complete lattice under $\preceq_k$.

Note that $\Pi_{\mathscr{P}}$ has been defined first in (Loyer and Straccia, 2003c) without recognizing that it characterizes supported models. However, it has been shown in (Loyer and Straccia, 2003c) that the least fixed-point under $\preceq_k$ coincides with the well-founded semantics, i.e. in our context, *the $\preceq_k$-least supported model of $\mathscr{P}$ is the well-founded semantics of $\mathscr{P}$.*

**Theorem 3.17** (*Loyer and Straccia 2003c*)
Consider a logic program $\mathscr{P}$. Then $WF(\mathscr{P}) = \mathrm{lfp}_{\preceq_k}(\Pi_{\mathscr{P}})$ $(WF(\mathscr{P}) = \mathrm{lfp}_{\preceq_k}(\tilde{\Pi}_{\mathscr{P}}))$ and stable models are fixed-points of $\Pi_{\mathscr{P}}$ $(\tilde{\Pi}_{\mathscr{P}})$.

**Example 3.18** (*running example cont.*)
Consider Table 3. Note that stable models are supported models, i.e. fixed-points of $\tilde{\Pi}_{\mathscr{P}}$ $(\Pi_{\mathscr{P}})$, and that the $\preceq_k$-least supported model coincides with the well-founded model. Additionally, $I_8$ and $I_9$ are fixed-points of $\tilde{\Pi}_{\mathscr{P}}$ $(\Pi_{\mathscr{P}})$ and not stable models. Thus, stable models are a proper subset of supported models.

Note that the above theorem is not surprising considering that the $\tilde{\Pi}_{\mathscr{P}}$ operator is quite similar to the $W_{\mathscr{P}}$ operator defined in Equation (7) for classical logic programs and interpretations. The above theorem essentially extends the relationship to general logic programs interpreted over bilattices. But, while for classical logical programs and total interpretations, $\tilde{\Pi}_{\mathscr{P}}(I)$ characterizes stable total models (as, $\tilde{\Pi}_{\mathscr{P}} = W_{\mathscr{P}}$), this is not true in the general case of interpretations over bilattices (e.g., see Table 3).

As highlighted in Examples 3.14 and 3.18, supported models are not specific enough to completely identify stable models: we must further refine the notion of supported models. Example 3.14 gives us a hint. For instance, consider the supported model $I_8$. As already noted, the support (in classical terms, the greatest unfounded set) of $I_8$ coincides with that of $I_5$, but for this support, i.e. $s_{\mathscr{P}}(I_5)$, $I_5$ is the $\preceq_k$-least informative cl-model, i.e. $I_5 \preceq_k I_8$. Similarly, for support $s_{\mathscr{P}}(I_6)$, $I_6$ is the $\preceq_k$-least informative cl-model, i.e. $I_6 \preceq_k I_9$. It appears clearly that some supported models contain knowledge that cannot be inferred from the program or from CWA. This may suggest partitioning supported models into sets of cl-models with a given support and then taking the least informative one to avoid that the supported models contain unexpected extra knowledge.

Formally, for a given interpretation $I$, we will consider the class of all models of $\mathscr{P} \oplus s_{\mathscr{P}}(I)$, i.e. interpretations which contain the knowledge entailed by $\mathscr{P}$ and the support $s_{\mathscr{P}}(I)$, and then take the $\preceq_k$-least model. If this $\preceq_k$-least model is $I$ itself then $I$ is a supported model of $\mathscr{P}$ deductively closed under support k-completion.

**Definition 3.19** (*model deductively closed under support k-completion*)
Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. Then $I$ is a *model deductively closed under support k-completion* of $P$ iff $I = \min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I)))$.

Note that by Lemma 2.16,

$$mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I)) = \{J : J = \Phi_{\mathscr{P}}(J) \oplus s_{\mathscr{P}}(I)\}. \tag{13}$$

Therefore, if $I$ is a model deductively closed under support k-completion then $I = \Phi_{\mathscr{P}}(I) \oplus s_{\mathscr{P}}(I)$, i.e. $I \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$. Therefore, by Theorem 3.15, any model deductively closed under support k-completion is also a supported model, i.e. $I \models_{cl} \mathscr{P}$ and $s_{\mathscr{P}}(I) \preceq_k I$.

Interestingly, models deductively closed under support k-completion have also a different, equivalent and quite suggestive characterization. In fact, from the definition it follows immediately that

$$\min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I))) \quad = \quad KK(\mathscr{P} \oplus s_{\mathscr{P}}(I)).$$

Table 4. *Running example cont.: models deductively closed under support k-completion of $\mathscr{P}$.*

| $I_i \models_{cl} \mathscr{P}$ | $I_i$ | | | $s_{\mathscr{P}}(I_i)$ | | | $U_{\mathscr{P}}(I_i)$ | $KK(\mathscr{P})$ | $WF(\mathscr{P})$ | stable models | supp. models | deductively closed models |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $p$ | $q$ | $r$ | $p$ | $q$ | $r$ | | | | | | |
| $I_1$ | $\perp$ | $\perp$ | $\perp$ | f | $\perp$ | $\perp$ | $\{p\}$ | • | | | | |
| $I_2$ | $\perp$ | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | | |
| $I_3$ | f | $\perp$ | $\perp$ | f | $\perp$ | $\perp$ | $\{p\}$ | | • | • | • | • |
| $I_4$ | f | f | t | f | f | $\perp$ | $\{p,q\}$ | | | • | • | • |
| $I_5$ | f | t | f | f | $\perp$ | f | $\{p,r\}$ | | | • | • | • |
| $I_6$ | f | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | • | • | • |
| $I_7$ | t | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | | |
| $I_8$ | $\top$ | t | f | f | $\perp$ | f | $\{p,r\}$ | | | | • | |
| $I_9$ | $\top$ | $\top$ | $\top$ | f | f | f | $\{p,q,r\}$ | | | | • | |

It then follows that

*Theorem 3.20*
Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. Then $I$ is a model deductively closed under support k-completion of $\mathscr{P}$ iff $I = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I))$.

That is, given an interpretation $I$ and logic program $\mathscr{P}$, among all cl-models of $\mathscr{P}$, we are looking for the $\preceq_k$-least cl-models deductively closed under support $k$-completion, i.e. models containing only the knowledge that can be inferred from $\mathscr{P}$ and from the safe part of CWA identified by its k-maximal safe interpretation.

*Example 3.21* (*running example cont.*)
Table 4 extends Table 3, by including models deductively closed under support k-completion. Note that now both $I_8$ and $I_9$ have been ruled out, as they are not minimal with respect to a given support, i.e. $I_8 \neq \min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I_8))) = \min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I_5))) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I_5)) = I_5$ and $I_9 \neq KK(\mathscr{P} \oplus s_{\mathscr{P}}(I_9)) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I_6)) = I_6$.

Finally, we can note that an immediate consequence operator characterizing models deductively closed under support k-completion can be derived immediately from Theorem 3.20, i.e. by relying on the operator $KK(\mathscr{P} \oplus s_{\mathscr{P}}(\cdot))$. In the following we present the operator $\Phi'_{\mathscr{P}}$, which coincides with $KK(\mathscr{P} \oplus s_{\mathscr{P}}(\cdot))$, i.e. $\Phi'_{\mathscr{P}}(I) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I))$ for any interpretation $I$, but does not require any, even intuitive, program transformation like $\mathscr{P} \oplus s_{\mathscr{P}}(\cdot)$. This may be important in the classical logic programming case where $\mathscr{P} \oplus s_{\mathscr{P}}(\cdot)$ is not easy to define (as $\oplus$ does not belong to the language of classical logic programs). Therefore, the set of models deductively closed under support k-completion coincides with the set of fixed-points of $\Phi'_{\mathscr{P}}$, which will be defined in terms of $\Phi_{\mathscr{P}}$ only.

Informally, given an interpretation $I$, $\Phi'_{\mathscr{P}}$ computes all the knowledge that can be inferred from the rules and the support of $\mathscr{P}$ w.r.t. $I$ without any other extra knowledge. Formally,

*Definition 3.22* (*immediate consequence operator* $\Phi'_{\mathscr{P}}$)

Consider a logic program $\mathscr{P}$ and an interpretation $I$. The operator $\Phi'_{\mathscr{P}}$ maps interpretations into interpretations and is defined as the limit of the sequence of interpretations $J_i^I$ defined as follows: for any $i \geqslant 0$,

$$
\begin{aligned}
J_0^I &= s_{\mathscr{P}}(I), \\
J_{i+1}^I &= \Phi_{\mathscr{P}}(J_i^I) \oplus J_i^I.
\end{aligned}
$$

In the following, with $J_i^I$ we indicate the $i$-th iteration of the immediate consequence operator $\Phi'_{\mathscr{P}}$, according to Definition 3.22.

Essentially, given the current knowledge expressed by $I$ about an intended model of $\mathscr{P}$, we compute first the support, $s_{\mathscr{P}}(I)$, and then cumulate all the implicit knowledge that can be inferred from $\mathscr{P}$, by starting from the support.

It is easy to note that the sequence $J_i^I$ is monotone non-decreasing under $\preceq_k$ and, thus has a limit. The following theorem follows directly from Theorems 2.14 and 3.11, and from the Knaster-Tarski theorem.

*Theorem 3.23*

$\Phi'_{\mathscr{P}}$ is monotone w.r.t. $\preceq_k$. Therefore, $\Phi'_{\mathscr{P}}$ has a least (and a greatest) fixed-point under $\preceq_k$.

Finally, note that

- by definition $\Phi'_{\mathscr{P}}(I) = \Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I)) \oplus \Phi'_{\mathscr{P}}(I)$, and thus $\Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I)) \preceq_k \Phi'_{\mathscr{P}}(I)$; and
- for fixed-points of $\Phi'_{\mathscr{P}}$ we have that $I = \Phi_{\mathscr{P}}(I) \oplus I$ and, thus, $\Phi_{\mathscr{P}}(I) \preceq_k I$.

Before proving the last theorem of this section, we need the following lemma.

*Lemma 3.24*

Let $\mathscr{P}$ be a logic program and let $I$ and $K$ be interpretations. If $K \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$ then $\Phi'_{\mathscr{P}}(I) \preceq_k K$.

*Proof*

Assume $K \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$, i.e. by Lemma 2.16, $K = \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(K) = \Phi_{\mathscr{P}}(K) \oplus s_{\mathscr{P}}(I)$. Therefore, $s_{\mathscr{P}}(I) \preceq_k K$. We show by induction on $i$ that $J_i^I \preceq_k K$ and, thus, at the limit $\Phi'_{\mathscr{P}}(I) \preceq_k K$.

(i) Case $i = 0$. By definition, $J_0^I = s_{\mathscr{P}}(I) \preceq_k K$.

(ii) Induction step: suppose $J_i^I \preceq_k K$. Then by assumption and by induction we have that $J_{i+1}^I = \Phi_{\mathscr{P}}(J_i^I) \oplus J_i^I \preceq_k \Phi_{\mathscr{P}}(K) \oplus K = \Phi_{\mathscr{P}}(K) \oplus \Phi_{\mathscr{P}}(K) \oplus s_{\mathscr{P}}(I) = \Phi_{\mathscr{P}}(K) \oplus s_{\mathscr{P}}(I) = K$, which concludes. □

The following concluding theorem characterizes the set of models deductively closed under support k-completion in terms of fixed-points of $\Phi'_{\mathscr{P}}$.

*Theorem 3.25*

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. Then $\Phi'_{\mathscr{P}}(I) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I))$.

*Proof*

The Kripke-Kleene model (for ease denoted $K$) of $\mathscr{P} \oplus s_{\mathscr{P}}(I)$ under $\preceq_k$, is the limit of the sequence

$$
\begin{aligned}
K_0 &= \mathtt{I}_\perp \,, \\
K_{i+1} &= \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(K_i) \,.
\end{aligned}
$$

As $K \models_{cl} \mathscr{P} \oplus s_{\mathscr{P}}(I)$, by Lemma 3.24, $\Phi'_{\mathscr{P}}(I) \preceq_k K$. Now we show that $K \preceq_k \Phi'_{\mathscr{P}}(I)$, by proving by induction on $i$ that $K_i \preceq_k \Phi'_{\mathscr{P}}(I)$ and, thus, at the limit $K \preceq_k \Phi'_{\mathscr{P}}(I)$.

(*i*) Case $i = 0$. We have $K_0 = \mathtt{I}_\perp \preceq_k \Phi'_{\mathscr{P}}(I)$.

(*ii*) Induction step: suppose $K_i \preceq_k \Phi'_{\mathscr{P}}(I)$. Then, by induction we have $K_{i+1} = \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(K_i) \preceq_k \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(\Phi'_{\mathscr{P}}(I))$. As $s_{\mathscr{P}}(I) \preceq_k \Phi'_{\mathscr{P}}(I)$, by Lemma 2.16 it follows that $K_{i+1} \preceq_k \Phi_{\mathscr{P} \oplus s_{\mathscr{P}}(I)}(\Phi'_{\mathscr{P}}(I)) = \Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I)) \oplus s_{\mathscr{P}}(I) \preceq_k \Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I)) \oplus \Phi'_{\mathscr{P}}(I) = \Phi'_{\mathscr{P}}(I)$, which concludes.    □

It follows immediately that

*Corollary 3.26*

An interpretation $I$ is a model deductively closed under support k-completion of $\mathscr{P}$ iff $I$ is a fixed-point of $\Phi'_{\mathscr{P}}$.

We will now state that *the set of stable models coincides with the set of models deductively closed under support k-completion*. This statement implies that our approach leads to an epistemic characterization of the family of stable models. It also evidences the role of CWA in logic programming. Indeed, CWA can be seen as the additional support of falsehood to be added cumulatively to the Kripke-Kleene semantics to define some more informative semantics: the well-founded and the stable model semantics. Moreover, it gives a new fixed-point characterization of that family. Our fixed-point characterization is based on $\Phi_{\mathscr{P}}$ only and neither requires any program transformation nor separation of positive and negative literals/information. The proof of the following stable model characterization theorem can be found in the appendix.

*Theorem 3.27 (stable model characterization)*

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. The following statements are equivalent:

1. $I$ is a stable model of $\mathscr{P}$;
2. $I$ is a model deductively closed under support k-completion of $\mathscr{P}$,
   i.e. $I = \min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I)))$;
3. $I = \Phi'_{\mathscr{P}}(I)$;
4. $I = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I))$.

Considering a classical logic program $\mathscr{P}$, a partial interpretation is a stable model of $\mathscr{P}$ if and only if it is deductively closed under its greatest unfounded set completion, i.e. if and only if it coincides with the limit of the sequence:

$$
\begin{aligned}
J_0^I &= \neg.U_{\mathscr{P}}(I) \,, \\
J_{i+1}^I &= \Phi_{\mathscr{P}}(J_i^I) \cup J_i^I \,.
\end{aligned}
$$

Finally, it is well-known that the least stable model of $\mathscr{P}$ w.r.t. $\preceq_k$ coincides with $\mathscr{P}$'s well-founded semantics. Therefore, our approach also provides new characterizations of the well-founded semantics of logic programs over bilattices. Together with Theorem 3.17, we have

*Corollary 3.28*
Let $\mathscr{P}$ be a logic program. The following statements are equivalent:

1. $I$ is the well-founded semantics of $\mathscr{P}$;
2. $I$ is the $\preceq_k$-least supported model of $\mathscr{P}$, i.e. the $\preceq_k$-least fixed-point of $\tilde{\Pi}_{\mathscr{P}}$;
3. $I$ is the $\preceq_k$-least model deductively closed under support k-completion of $\mathscr{P}$, i.e. the $\preceq_k$-least fixed-point of $\Phi'_{\mathscr{P}}$.

Therefore, the well-founded semantics can be characterized by means of the notion of supported models only. Additionally, we now also know why $\tilde{\Pi}_{\mathscr{P}}$ characterizes the well-founded model, while fails in characterizing stable models. Indeed, from $I = \tilde{\Pi}_{\mathscr{P}}(I)$ it follows that $I$ is a model of $\mathscr{P} \oplus s_{\mathscr{P}}(I)$, which does not guarantee that $I$ is the $\preceq_k$-least cl-model of $\mathscr{P} \oplus s_{\mathscr{P}}(I)$ (see Example 3.21). Thus, $I$ does not satisfy Theorem 3.20. If $I$ is the $\preceq_k$-least fixed-point of $\tilde{\Pi}_{\mathscr{P}}$, then $I$ is both a cl-model of $\mathscr{P} \oplus s_{\mathscr{P}}(I)$ and $\preceq_k$-least. Therefore, the $\preceq_k$-least supported model is always a model deductively closed under support k-completion as well and, thus a stable model.

The following concluding example shows the various ways of computing the well-founded semantics, according to the operators discussed in this study: $\Psi'_{\mathscr{P}}$ and $\Phi'_{\mathscr{P}}$. But, rather than relying on $\mathscr{FOUR}$ as truth space, as we did in our running example, we consider the bilattice of intervals over the unit $[0,1]$, used frequently for reasoning under uncertainty.

*Example 3.29*
Let us consider the bilattice of intervals $\langle [0,1] \times [0,1], \preceq_t, \preceq_k \rangle$ introduced in Example 3.5. Consider the following logic program $\mathscr{P}$,

$$
\begin{aligned}
A &\leftarrow A \vee B \\
B &\leftarrow (\neg C \wedge A) \vee \langle 0.3, 0.5 \rangle \\
C &\leftarrow \neg B \vee \langle 0.2, 0.4 \rangle
\end{aligned}
$$

The table below shows the computation of the Kripke-Kleene semantics of $\mathscr{P}$, $KK(\mathscr{P})$, as $\preceq_k$-least fixed-point of $\Phi_{\mathscr{P}}$.

| $A$ | $B$ | $C$ | $K_i$ |
|---|---|---|---|
| $\langle 0,1 \rangle$ | $\langle 0,1 \rangle$ | $\langle 0,1 \rangle$ | $K_0$ |
| $\langle 0,1 \rangle$ | $\langle 0.3,1 \rangle$ | $\langle 0.2,1 \rangle$ | $K_1$ |
| $\langle 0.3,1 \rangle$ | $\langle 0.3,0.8 \rangle$ | $\langle 0.2,0.7 \rangle$ | $K_2$ |
| $\langle 0.3,1 \rangle$ | $\langle 0.3,0.8 \rangle$ | $\langle 0.2,0.7 \rangle$ | $K_3 = K_2 = KK(\mathscr{P})$ |

Note that knowledge increases during the computation as the intervals become more precise, i.e. $K_i \preceq_k K_{i+1}$.

The following table shows us the computation of the well-founded semantics of $\mathscr{P}$, $WF(\mathscr{P})$, as $\preceq_k$-least fixed-point of $\Psi'_{\mathscr{P}}$.

| $\dfrac{W_j}{v_i}$ | $A$ | $B$ | $C$ | $A$ | $B$ | $C$ | $W_j$ |
|---|---|---|---|---|---|---|---|
| $v_0^{W_0}$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,1\rangle$ | $\langle 0,1\rangle$ | $\langle 0,1\rangle$ | $W_0$ |
| $v_1^{W_0}$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| $v_2^{W_0}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| $v_3^{W_0}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| $v_0^{W_1}$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0,1\rangle$ | $W_1$ |
| $v_1^{W_1}$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| $v_2^{W_1}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| $v_3^{W_1}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| $v_0^{W_2}$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $W_2$ |
| $v_1^{W_2}$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| $v_2^{W_2}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| $v_3^{W_2}$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | | | | |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $W_3 = W_2 = WF(\mathscr{P})$ |

Note that $W_i \preceq_k W_{i+1}$ and $KK(\mathscr{P}) \preceq_k WF(\mathscr{P})$, as expected. We conclude this example by showing the computation of the well-founded semantics of $\mathscr{P}$, as $\preceq_k$-least fixed-point of $\Phi'_{\mathscr{P}}$.

| $F_i^{I_n}$ | $A$ | $B$ | $C$ | $A$ | $B$ | $C$ | $I_n / J_j^{I_n}$ |
|---|---|---|---|---|---|---|---|
| $F_0^{I_0}$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,1\rangle$ | $\langle 0,1\rangle$ | $\langle 0,1\rangle$ | $I_0$ |
| $F_1^{I_0}$ | $\langle 0,0\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| $F_2^{I_0}$ | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| $F_3^{I_0}$ | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,1\rangle$ | | | | |
| | | | | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,1\rangle$ | $J_0^{I_0} = s_{\mathscr{P}}(I_0)$ |
| | | | | $\langle 0,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,1\rangle$ | $J_1^{I_0}$ |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $J_2^{I_0}$ |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $J_3^{I_0}$ |
| $F_0^{I_1}$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0,0\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $I_1$ |
| $F_1^{I_1}$ | $\langle 0,0\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,0.7\rangle$ | | | | |
| $F_2^{I_1}$ | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,0.7\rangle$ | | | | |
| $F_3^{I_1}$ | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,0.7\rangle$ | | | | |
| | | | | $\langle 0,0.5\rangle$ | $\langle 0,0.5\rangle$ | $\langle 0,0.7\rangle$ | $J_0^{I_1} = s_{\mathscr{P}}(I_1)$ |
| | | | | $\langle 0,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $J_1^{I_1}$ |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $J_2^{I_1}$ |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $J_3^{I_1}$ |
| | | | | $\langle 0.3,0.5\rangle$ | $\langle 0.3,0.5\rangle$ | $\langle 0.5,0.7\rangle$ | $I_2 = I_1 = WF(\mathscr{P})$ |

Table 5. *Well-founded semantics characterization: from classical logic to bilattices.*

| 1. | $I$ is the well-founded semantics of $\mathscr{P}$ | |
|---|---|---|
| | Classical logic $\{\mathtt{f}, \bot, \mathtt{t}\}$ | **Bilattices** |
| 2. $\preceq_k$-least $I$ s.t. | $I = W_{\mathscr{P}}(I) = T_{\mathscr{P}}(I) \cup \neg.U_{\mathscr{P}}(I)$ | $\mathbf{I} = \tilde{\mathbf{\Pi}}_{\mathscr{P}}(\mathbf{I}) = \mathbf{\Phi}_{\mathscr{P}}(\mathbf{I}) \oplus \mathbf{s}_{\mathscr{P}}(\mathbf{I})$ |
| 3. $\preceq_k$-least model $I$ s.t. | $\neg.U_{\mathscr{P}}(I) \subseteq I$ | $\mathbf{s}_{\mathscr{P}}(\mathbf{I}) \preceq_{\mathbf{k}} \mathbf{I}$ |

Note how the knowledge about falsehood increases as our approximation to the intended model increases, i.e. $s_{\mathscr{P}}(I_i) \preceq_k s_{\mathscr{P}}(I_{i+1})$, while the degree of truth decreases ($s_{\mathscr{P}}(I_{i+1}) \preceq_t s_{\mathscr{P}}(I_i)$). Furthermore, note that $WF(\mathscr{P}) \models_{cl} \mathscr{P}$ and $s_{\mathscr{P}}(WF(\mathscr{P})) \preceq_k WF(\mathscr{P})$, i.e. $WF(\mathscr{P})$ is a supported model of $\mathscr{P}$, compliant to Corollary 3.28.

## 4 Conclusions

Stable model semantics has become a well-established and accepted approach to the management of (non-monotonic) negation in logic programs. In this study we have presented an alternative formulation to the Gelfond-Lifschitz transformation, which has widely been used to formulate stable model semantics. Our approach is purely based on algebraic and semantical aspects of informative monotone operators over bilattices. In this sense, we talk about epistemological foundation of the stable model semantics. Our considerations are based on the fact that we regard the closed world assumption as an additional source of falsehood and identify with the *support* the amount/degree of falsehood carried on by the closed world assumption. The support is the generalization of the notion of the greatest unfounded set for classical logic programs to the context of bilattices. The support is then used to complete the well-known Kripke-Kleene semantics of logic programs. In particular,

1. with respect to well-founded semantics, we have generalized both the fixed-point characterization of the well-founded semantics of van Gelder *et al.* (1991) to bilattices (Point 2 in Table 5) and its model-theoretic characterization (Point 3 in Table 5, e.g., see Leone *et al.* (1997)).
2. concerning stable model semantics, we have shown that

$$I \in stable(\mathscr{P}) \text{ iff } I = \min_{\preceq_k}(mod_{cl}(\mathscr{P} \oplus s_{\mathscr{P}}(I))) = KK(\mathscr{P} \oplus s_{\mathscr{P}}(I)) = \Phi'_{\mathscr{P}}(I) ,$$

while previously stable models have been characterized by $I \in stable(\mathscr{P})$ iff $I = \min_{\preceq_t}(mod(\mathscr{P}^I))$.

Our result indicates that the support may be seen as the added-value to the Kripke-Kleene semantics and evidences the role of CWA in the well-founded and stable model semantics. It also shows that a separation of positive and negative information is nor necessary (as required by the Gelfond-Lifschitz transform), nor is any program transformation required.

As our approach is rather general and abstracts from the underlying logical formalism (in our case logic programs), it may be applied to other contexts as well.

## A Proof of Theorem 3.27

This part is devoted to the proof of Theorem 3.27. It relies on the following intermediary results. We start by providing lemmas to show that fixed-points of $\Phi'_{\mathscr{P}}$ are stable models.

*Lemma A.1*
If $I \preceq_t J$ and $J \preceq_k I$, then $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, J)$, for any interpretation $x$.

*Proof*
Using the antimonotonicity of $\Psi_{\mathscr{P}}$ w.r.t. $\preceq_t$ for its second argument, we have $\mathtt{I_f} \preceq_t \Psi_{\mathscr{P}}(x, J) \preceq_t \Psi_{\mathscr{P}}(x, I)$. From Lemma 2.2, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, I) \preceq_k \Psi_{\mathscr{P}}(x, J)$. Using the interlacing conditions, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, I) \preceq_k \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, J)$. Now, using the monotonicity of $\Psi_{\mathscr{P}}$ w.r.t. $\preceq_k$ and the interlacing conditions, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, J) \preceq_k \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, I)$. It results that $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(x, J)$. $\square$

Similarly, we have

*Lemma A.2*
If $J \preceq_t I$ and $J \preceq_k I$, then $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I, x) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(J, x)$, for any interpretation $x$.

*Proof*
Using the monotonicity of $\Psi_{\mathscr{P}}$ w.r.t. $\preceq_t$ for its first argument, we have $\mathtt{I_f} \preceq_t \Psi_{\mathscr{P}}(J, x) \preceq_t \Psi_{\mathscr{P}}(I, x)$. From Lemma 2.2, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I, x) \preceq_k \Psi_{\mathscr{P}}(J, x)$. Using the interlacing conditions, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I, x) \preceq_k \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(J, x)$. Now, using the monotonicity of $\Psi_{\mathscr{P}}$ w.r.t. $\preceq_k$ and the interlacing conditions, we have $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(J, x) \preceq_k \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I, x)$. It results that $\mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I, x) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(J, x)$. $\square$

*Lemma A.3*
If $I = \Phi_{\mathscr{P}}(I)$ then $F_i^I \preceq_t s_{\mathscr{P}}(I) \preceq_t I$, for all $i$.

*Proof*
By Theorem 3.10, the sequence $F_i^I$ is monotone non-decreasing under $\preceq_t$ and $F_i^I \preceq_t s_{\mathscr{P}}(I)$. Now, we show by induction on $i$ that $F_i^I \preceq_t I$ and, thus, at the limit $s_{\mathscr{P}}(I) \preceq_t I$.
  (i) Case $i = 0$. $F_0^I = \mathtt{I_f} \preceq_t I$.
  (ii) Induction step: let us assume that $F_i^I \preceq_t I$ holds. By Lemma 2.3, $F_i^I \preceq_t F_i^I \oplus I \preceq_t I$ follows. We also have $I \preceq_k F_i^I \oplus I$ and $F_i^I \preceq_k F_i^I \oplus I$. It follows from Lemma A.1 and Lemma A.2 that $F_{i+1}^I = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I \oplus I, F_i^I \oplus I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I, I)$. By induction $F_i^I \preceq_t I$, so from $I = \Phi_{\mathscr{P}}(I)$, $F_{i+1}^I = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I, I) \preceq_t \Psi_{\mathscr{P}}(F_i^I, I) \preceq_t \Psi_{\mathscr{P}}(I, I) = \Phi_{\mathscr{P}}(I) = I$ follows. $\square$

*Lemma A.4*
If $I = \Phi_{\mathscr{P}}(I)$ then for any $i$, $s_{\mathscr{P}}(I) \preceq_k F_i^I \preceq_k v_i^I$ and, thus, at the limit $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I)$.

*Proof*

By Theorem 3.10, $s_{\mathscr{P}}(I) \preceq_k F_i^I$, for all $i$. We know that $v_i^I$ converges to $\Psi'_{\mathscr{P}}(I)$. We show by induction on $i$ that $F_i^I \preceq_k v_i^I$. Therefore, at the limit $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I)$.

(*i*) Case $i = 0$. $F_0^I = \mathtt{I_f} \preceq_k \mathtt{I_f} = v_0^I$.

(*ii*) Induction step: assume that $F_i^I \preceq_k v_i^I$. By definition, $F_{i+1}^I = \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus F_i^I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(I \oplus F_i^I, I \oplus F_i^I)$. By Lemma A.3, $F_i^I \preceq_t I$. By Lemma 2.3, $F_i^I \preceq_t F_i^I \oplus I \preceq_t I$ follows. We also have $I \preceq_k F_i^I \oplus I$ and $F_i^I \preceq_k F_i^I \oplus I$. It follows from Lemma A.1 and Lemma A.2 that $F_{i+1}^I = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I \oplus I, F_i^I \oplus I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I, I)$. By the induction hypothesis we know that $F_i^I \preceq_k v_i^I$ for any $n$. Therefore, $F_{i+1}^I \preceq_k \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(v_i^I, I) \preceq_k \Psi_{\mathscr{P}}(v_i^I, I) = v_{i+1}^I$ follows, which concludes. $\quad\square$

*Lemma A.5*

Let $\mathscr{P}$ and $I$ be a logic program and an interpretation, respectively. If $I$ is a supported model then $s_{\mathscr{P}}(I) = \mathtt{I_f} \otimes I$.

*Proof*

By Equation 10 and Theorem 3.15, $s_{\mathscr{P}}(I) = \mathtt{I_f} \otimes \Phi_{\mathscr{P}}(I \oplus s_{\mathscr{P}}(I)) = \mathtt{I_f} \otimes I$. $\quad\square$

*Lemma A.6*

If $I = \Phi'_{\mathscr{P}}(I)$ then we have:

1. $s_{\mathscr{P}}(I) \preceq_t \Psi'_{\mathscr{P}}(I) \preceq_t I$; and
2. $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I) \preceq_k I$.

*Proof*

By Corollary 3.26 and by Lemma A.5, $s_{\mathscr{P}}(I) = \mathtt{I_f} \otimes I$ and $I = \Phi_{\mathscr{P}}(I)$. From Lemma A.4, $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I)$. By definition of $\Psi'_{\mathscr{P}}$, $\Psi'_{\mathscr{P}}(I) = \mathrm{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathscr{P}}(x, I))$. But, $I = \Phi_{\mathscr{P}}(I) = \Psi_{\mathscr{P}}(I, I)$, thus $\Psi'_{\mathscr{P}}(I) \preceq_t I$.

Now we show by induction on $i$, that $F_i^I \preceq_t v_i^I$. Therefore, at the limit, $s_{\mathscr{P}}(I) \preceq_t \Psi'_{\mathscr{P}}(I)$ and, thus, $s_{\mathscr{P}}(I) \preceq_t \Psi'_{\mathscr{P}}(I) \preceq_t I$ hold.

(*i*) Case $i = 0$. $F_0^I = \mathtt{I_f} \preceq_t \mathtt{I_f} = v_0^I$.

(*ii*) Induction step: let us assume that $F_i^I \preceq_t v_i^I$ holds. From Lemma A.3, we have $F_i^I \preceq_t I$ and, thus, by Lemma 2.3, $F_i^I \preceq_t F_i^I \oplus I \preceq_t I$ follows. We also have $I \preceq_k F_i^I \oplus I$ and $F_i^I \preceq_k F_i^I \oplus I$. Then, from Lemma A.1 and Lemma A.2, $F_{i+1}^I = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I \oplus I, F_i^I \oplus I) = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I, I)$. By induction $F_i^I \preceq_t v_i^I$, so by Lemma 2.4 we have $F_{i+1}^I = \mathtt{I_f} \otimes \Psi_{\mathscr{P}}(F_i^I, I) \preceq_t \Psi_{\mathscr{P}}(F_i^I, I) \preceq_t \Psi_{\mathscr{P}}(v_i^I, I) = v_{i+1}^I$, which concludes.

Finally, from $s_{\mathscr{P}}(I) \preceq_t \Psi'_{\mathscr{P}}(I) \preceq_t I$ and by Lemma 2.2 we have $\Psi'_{\mathscr{P}}(I) \preceq_k I \oplus s_{\mathscr{P}}(I) = I$, so $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I) \preceq_k I$. $\quad\square$

Now we are ready to show that fixed-points of $\Phi'_{\mathscr{P}}$ are stable models.

*Theorem A.7*

Every fixed-point of $\Phi'_{\mathscr{P}}$ is a stable model of $\mathscr{P}$.

*Proof*

Assume $I = \Phi'_{\mathscr{P}}(I)$. Let us show that $I = \Psi'_{\mathscr{P}}(I)$. From Lemma A.6, we know that $\Psi'_{\mathscr{P}}(I) \preceq_k I$. Now, let us show by induction on $i$ that $J_i^I \preceq_k \Psi'_{\mathscr{P}}(I)$. Therefore, at the limit $I = \Phi'_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I)$ and, thus, $I = \Psi'_{\mathscr{P}}(I)$.

(i) Case $i = 0$. $J_0^I = s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I)$, by Lemma A.6.

(ii) Induction step: let us assume that $J_i^I \preceq_k \Psi'_{\mathscr{P}}(I)$ holds. By definition, $J_{i+1}^I = \Phi_{\mathscr{P}}(J_i^I) \oplus J_i^I$. By induction $J_i^I \preceq_k \Psi'_{\mathscr{P}}(I)$. Therefore, $J_{i+1}^I \preceq_k \Phi_{\mathscr{P}}(\Psi'_{\mathscr{P}}(I)) \oplus \Psi'_{\mathscr{P}}(I)$. But, by Lemma A.6, $\Psi'_{\mathscr{P}}(I) \preceq_k I$, so $\Phi_{\mathscr{P}}(\Psi'_{\mathscr{P}}(I)) = \Psi_{\mathscr{P}}(\Psi'_{\mathscr{P}}(I), \Psi'_{\mathscr{P}}(I)) \preceq_k \Psi_{\mathscr{P}}(\Psi'_{\mathscr{P}}(I), I) = \Psi'_{\mathscr{P}}(I)$. Therefore, $J_{i+1}^I \preceq_k \Psi'_{\mathscr{P}}(I)$.  $\square$

The following lemmas are needed to show the converse, i.e. that stable models are fixed-points of $\Phi'_{\mathscr{P}}$.

*Lemma A.8*
If $I = \Psi'_{\mathscr{P}}(I)$ then we have:

1. $s_{\mathscr{P}}(I) \preceq_k I$;
2. $\Phi'_{\mathscr{P}}(I) \preceq_k I$;
3. $\Phi'_{\mathscr{P}}(I) \preceq_t I$.

*Proof*
Assume $I = \Psi'_{\mathscr{P}}(I)$. By Theorem 2.20, $I = \Phi_{\mathscr{P}}(I)$. By Lemma A.4, $s_{\mathscr{P}}(I) \preceq_k \Psi'_{\mathscr{P}}(I) = I$, which completes Point 1..

Now, we show by induction on $i$ that, $J_i^I \preceq_k I$ and $J_i^I \preceq_t I$ and, thus, at the limit $\Phi'_{\mathscr{P}}(I) \preceq_k I$ and $\Phi'_{\mathscr{P}}(I) \preceq_t I$ hold.

(i) Case $i = 0$. By Point 1., $J_0^I = s_{\mathscr{P}}(I) \preceq_k I$, while $J_0^I = s_{\mathscr{P}}(I) \preceq_t I$, by Lemma A.3.

(ii) Induction step: let us assume that $J_i^I \preceq_k I$ and $J_i^I \preceq_t I$ hold. By definition, $J_{i+1}^I = \Phi_{\mathscr{P}}(J_i^I) \oplus J_i^I$. By induction $J_i^I \preceq_k I$, thus $J_{i+1}^I \preceq_k \Phi_{\mathscr{P}}(I) \oplus I = I \oplus I = I$, which completes Point 2. From $J_i^I \preceq_k I$, $\Phi_{\mathscr{P}}(J_i^I) \preceq_k \Phi_{\mathscr{P}}(I) = I$ follows. By induction we have $J_i^I \preceq_t I$, thus $J_{i+1}^I \preceq_t \Phi_{\mathscr{P}}(J_i^I) \oplus I = I$, which completes Point 3.  $\square$

*Lemma A.9*
If $I = \Psi'_{\mathscr{P}}(I)$ then $I \preceq_t \Phi'_{\mathscr{P}}(I)$.

*Proof*
Assume $I = \Psi'_{\mathscr{P}}(I)$. By Theorem 2.20, $I = \Phi_{\mathscr{P}}(I)$. By Lemma A.3 and Lemma A.8, $s_{\mathscr{P}}(I) \preceq_k I$ and $s_{\mathscr{P}}(I) \preceq_t I$, so by Lemma 2.7, $s_{\mathscr{P}}(I) = s_{\mathscr{P}}(I) \otimes \mathtt{I_f} = I \otimes \mathtt{I_f}$.

Now, we show by induction on $i$, that $v_i^I \preceq_t \Phi'_{\mathscr{P}}(I)$. Therefore, at the limit, $I = \Psi'_{\mathscr{P}}(I) \preceq_t \Phi'_{\mathscr{P}}(I)$.

(i) Case $i = 0$. $v_0^I = \mathtt{I_f} \preceq_t \Phi'_{\mathscr{P}}(I)$.

(ii) Induction step: let us assume that $v_i^I \preceq_t \Phi'_{\mathscr{P}}(I)$ holds. By definition and by the induction hypothesis, $v_{i+1}^I = \Psi_{\mathscr{P}}(v_i^I, I) \preceq_t \Psi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I), I)$. By Lemma A.8, $\Phi'_{\mathscr{P}}(I) \preceq_t I$. Therefore, since $\Psi_{\mathscr{P}}$ is antitone in the second argument under $\preceq_t$, $v_{i+1}^I \preceq_t \Psi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I), \Phi'_{\mathscr{P}}(I)) = \Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I))$. It follows that $v_i^I \oplus v_{i+1}^I \preceq_t \Phi_{\mathscr{P}}(\Phi'_{\mathscr{P}}(I)) \oplus \Phi'_{\mathscr{P}}(I) = \Phi'_{\mathscr{P}}(I)$. By Lemma 2.5, (by assuming, $x = v_i^I, z = v_{i+1}^I, y = \Phi'_{\mathscr{P}}(I)$), $v_{i+1}^I \preceq_k \Phi'_{\mathscr{P}}(I) \oplus \mathtt{I_f}$ follows. By Lemma A.8, both $\Phi'_{\mathscr{P}}(I) \preceq_t I$ and $\Phi'_{\mathscr{P}}(I) \preceq_k I$ hold. Therefore, by Lemma 2.7, $\Phi'_{\mathscr{P}}(I) \otimes \mathtt{I_f} = I \otimes \mathtt{I_f} = s_{\mathscr{P}}(I)$. From Lemma A.4, $\Phi'_{\mathscr{P}}(I) \otimes \mathtt{I_f} = s_{\mathscr{P}}(I) \preceq_k v_{i+1}^I \preceq_k \Phi'_{\mathscr{P}}(I) \oplus \mathtt{I_f}$. Therefore, by Lemma 2.6, it follows that $v_{i+1}^I \preceq_t \Phi'_{\mathscr{P}}(I)$, which concludes the proof.  $\square$

We can now prove that every stable model is indeed a fixed-point of $\Phi'_{\mathscr{P}}$, which concludes the characterization of stable models on bilattices.

*Theorem A.10*
Every stable model of $\mathscr{P}$ is a fixed-point of $\Phi'_{\mathscr{P}}$.

*Proof*
Assume $I = \Psi'_{\mathscr{P}}(I)$. By Lemma A.8, $\Phi'_{\mathscr{P}}(I) \preceq_t I$, while by Lemma A.9, $I \preceq_t \Phi'_{\mathscr{P}}(I)$. So $I = \Phi'_{\mathscr{P}}(I)$. $\square$

Finally, Theorem 3.27 follows directly from Theorems A.7, A.10, 3.20 and Corollary 3.26.

# References

ALCANTÂRA, J., DAMÁSIO, C. V. AND PEREIRA, L. M. 2002. Paraconsistent logic programs. *Proc. of the 8th European Conference on Logics in Artificial Intelligence (JELIA-02)*. Number 2424 in Lecture Notes in Computer Science. Springer-Verlag, Cosenza, Italy, 345–356.

ANDERSON, A. R. AND BELNAP, N. D. 1975. *Entailment – the logic of relevance and necessity*. Princeton University Press, Princeton, NJ.

ARIELI, O. 2002. Paraconsistent declarative semantics for extended logic programs. *Annals of Mathematics and Artificial Intelligence 36,* 4, 381–417.

ARIELI, O. AND AVRON, A. 1996. Reasoning with logical bilattices. *Journal of Logic, Language and Information 5,* 1, 25–63.

ARIELI, O. AND AVRON, A. 1998. The value of the four values. *Artificial Intelligence Journal 102,* 1, 97–141.

AVRON, A. 1996. The structure of interlaced bilattices. *Journal of Mathematical Structures in Computer Science 6*, 287–299.

BELNAP, N. D. 1977. A useful four-valued logic. In *Modern uses of multiple-valued logic*, G. Epstein and J. M. Dunn, Eds. Reidel, Dordrecht, NL, 5–37.

BLAIR, H. AND SUBRAHMANIAN, V. S. 1989. Paraconsistent logic programming. *Theoretical Computer Science 68*, 135–154.

CLARK, K. 1978. Negation as failure. In *Logic and data bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York, NY, 293–322.

DAMÁSIO, C. V. AND PEREIRA, L. M. 1998. A survey of paraconsistent semantics for logic programs. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, D. Gabbay and P. Smets, Eds. Kluwer, 241–320.

DAMÁSIO, C. V. AND PEREIRA, L. M. 2001. Antitonic logic programs. *Proceedings of the 6th European Conference on logic programming and Nonmonotonic Reasoning (LPNMR-01)*. Number 2173 in Lecture Notes in Computer Science. Springer-Verlag.

DENECKER, M. 1998. The well-founded semantics is the principle of inductive definition. *Logics in Artificial Intelligence, Proceedings of JELIA-98*, J. Dix, L. Farinos del Cerro, and U. Furbach, Eds. Number 1489 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1–16.

DENECKER, M., BRUYNOOGHE, M. AND MAREK, V. 2001. Logic programming revisited: logic programs as inductive definitions. *ACM Transactions on Computational Logic (TOCL) 2,* 4, 623–654.

DENECKER, M., MAREK, V. W. AND TRUSZCZYŃSKI, M. 1999. Approximating operators, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. *NFS-workshop on Logic-based Artificial Intelligence*, J. Minker, Ed. 1–26.

DENECKER, M., MAREK, V. W. AND TRUSZCZYŃSKI, M. 2002. Ultimate approximations in nonmonotonic knowledge representation systems. *Principles of Knowledge Representation*

*and Reasoning: Proceedings of the 8th International Conference*, D. Fensel, F. Giunchiglia, D. McGuinness, and M. Williams, Eds. Morgan Kaufmann, 177–188.

DENECKER, M., MAREK, V. W. AND TRUSZCZYŃSKI, M. 2003. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence Journal 143*, 79–122.

DUNN, J. M. 1976. Intuitive semantics for first-degree entailments and coupled trees. *Philosophical Studies 29*, 149–168.

DUNN, J. M. 1986. Relevance logic and entailment. *Handbook of Philosophical Logic*, D. M. Gabbay and F. Guenthner, Eds. Vol. 3. Reidel, Dordrecht, NL, 117–224.

EMDEN, M. H. V. AND KOWALSKI, R. A. 1976. The semantics of predicate logic as a programming language. *Journal of the ACM (JACM) 23*, 4, 733–742.

FITTING, M. 1985. A Kripke-Kleene-semantics for general logic programs. *Journal of Logic Programming 2*, 295–312.

FITTING, M. 1991. Bilattices and the semantics of logic programming. *Journal of Logic Programming 11*, 91–116.

FITTING, M. 1992. Kleene's logic, generalized. *Journal of Logic and Computation 1*, 6, 797–810.

FITTING, M. C. 1993. The family of stable models. *Journal of Logic Programming 17*, 197–225.

FITTING, M. C. 2002. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science 21*, 3, 25–51.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. *Proceedings of the 5th International Conference on Logic Programming*, R. A. Kowalski and K. Bowen, Eds. The MIT Press, Cambridge, Massachusetts, 1070–1080.

GELFOND, M. AND LIFSCHITZ, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9*, 3/4, 365–386.

GINSBERG, M. L. 1988. Multi-valued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence 4*, 265–316.

HERRE, H. AND WAGNER, G. 1997. Stable models are generated by a stable chain. *Journal of Logic Programming 30*, 2, 165–177.

KUNEN, K. 1987. Negation in logic programming. *Journal of Logic Programming 4*, 4, 289–308.

LEONE, N., RULLO, P. AND SCARCELLO, F. 1997. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and Computation 135*, 2, 69–112.

LEVESQUE, H. J. 1984. A logic of implicit and explicit belief. *Proc. of the 3th Nat. Conf. on Artificial Intelligence (AAAI-84)*. Austin, TX, 198–202.

LEVESQUE, H. J. 1988. Logic and the complexity of reasoning. *Journal of Philosophical Logic 17*, 355–389.

LLOYD, J. W. 1987. *Foundations of Logic Programming*. Springer, Heidelberg, RG.

LOYER, Y. AND STRACCIA, U. 2002a. Uncertainty and partial non-uniform assumptions in parametric deductive databases. *Proc. of the 8th European Conference on Logics in Artificial Intelligence (JELIA-02)*. Number 2424 in Lecture Notes in Computer Science. Springer-Verlag, Cosenza, Italy, 271–282.

LOYER, Y. AND STRACCIA, U. 2002b. The well-founded semantics in normal logic programs with uncertainty. *Proc. of the 6th International Symposium on Functional and Logic Programming (FLOPS-2002)*. Number 2441 in Lecture Notes in Computer Science. Springer-Verlag, Aizu, Japan, 152–166.

LOYER, Y. AND STRACCIA, U. 2003a. The approximate well-founded semantics for logic programs with uncertainty. *28th International Symposium on Mathematical Foundations of Computer Science (MFCS-2003)*. Number 2747 in Lecture Notes in Computer Science. Springer-Verlag, Bratislava, Slovak Republic, 541–550.

LOYER, Y. AND STRACCIA, U. 2003b. Default knowledge in logic programs with uncertainty. *Proc. of the 19th Int. Conf. on Logic Programming (ICLP-03)*. Number 2916 in Lecture Notes in Computer Science. Springer Verlag, Mumbai, India, 466–480.

LOYER, Y. AND STRACCIA, U. 2003c. The well-founded semantics of logic programs over bilattices: an alternative characterisation. Technical Report ISTI-2003-TR-05, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy.

LUKASIEWICZ, T. 2001. Fixpoint characterizations for many-valued disjunctive logic programs with probabilistic semantics. *In Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-01)*. Number 2173 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 336–350.

MOORE, R. C. 1984. Possible-world semantics for autoepistemic logic. *Proceedings of the 1st International Workshop on Nonmonotonic Reasoning*. New Paltz, NY, 344–354.

NG, R. AND SUBRAHMANIAN, V. 1991. Stable model semantics for probabilistic deductive databases. *Proc. of the 6th Int. Sym. on Methodologies for Intelligent Systems (ISMIS-91)*, Z. W. Ras and M. Zemenkova, Eds. Number 542 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 163–171.

PRZYMUSINSKI, T. C. 1990a. Extended stable semantics for normal and disjunctive programs. In *Proceedings of the 7th International Conference on Logic Programming*, D. H. D. Warren and P. Szeredi, Eds. MIT Press, 459–477.

PRZYMUSINSKI, T. C. 1990b. Stationary semantics for disjunctive logic programs and deductive databases. *Logic Programming, Proceedings of the 1990 North American Conference*, S. Debray and H. Hermenegildo, Eds. MIT Press, 40–59.

PRZYMUSINSKI, T. C. 1990c. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae 13,* 4, 445–463.

REITER, R. 1978. On closed world data bases. *Logic and data bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York, NY, 55–76.

REITER, R. 1980. A logic for default reasoning. *Artificial Intelligence 13*, 81–132.

TARSKI, A. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics* 5, 285–309.

VAN GELDER, A. 1989. The alternating fixpoint of logic programs with negation. *Proc. of the 8th ACM SIGACT SIGMOD Sym. on Principles of Database Systems (PODS-89)*. 1–10.

VAN GELDER, A., ROSS, K. A. AND SCHLIMPF, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 620–650.