

# The Well-Founded Semantics of Logic Programs over Bilattices: an Alternative Characterisation

Yann Loyer <sup>a</sup>

<sup>a</sup> *Laboratoire PRiSM, Université de Versailles Saint Quentin  
45 Avenue des Etats-Unis, 78035 Versailles FRANCE*

Umberto Straccia <sup>b</sup>

<sup>b</sup> *I.S.T.I. - C.N.R.  
Via G. Moruzzi, 1 56124 Pisa (PI) ITALY*

---

## Abstract

The well-founded semantics is one of the most widely studied and used semantics of logic programs with negation. It is well-known that this semantics can be defined in terms of the “least” (according to the knowledge order) stable-model semantics, where this latter is based on the Gelfond-Lifschitz transformation. Fitting has generalized the Gelfond-Lifschitz transformation to the case where the underlying space of truth is that of bilattices and, thus, has extended the well-founded and the stable model semantics to bilattices. The main idea of the transformation is to separate the roles of positive and negative information, which, on the other hand, avoids the kind of symmetry and natural management of negation that is present in the well-known Kripke-Kleene semantics of logic programs with negation. In this paper, we show that this separation is not necessary. In particular, we show that the well-founded semantics over bilattices can be defined as an extension, based on the closed world assumption, of the Kripke-Kleene semantics. In particular, we define an immediate consequence operator  $\Pi_{\mathcal{P}}$ , which relies on Kripke-Kleene’s  $\Phi_{\mathcal{P}}$  operator only, whose least fixed-point according to the knowledge order coincides with the well-founded semantics over bilattices. As a consequence, we neither require any separation of roles of positive and negative information nor any program transformation, but rely on a natural management of negation only, making clearer the role of the closed world assumption in the well-founded semantics.

### ACM Categories and Subject Descriptors:

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods - *Representation languages*

*Key words:* Logic programs, Bilattices, Well-founded semantics

---

## 1 Introduction

One of the most important problems of logic programming consists in defining the intended meaning or semantics of any given logic program. Classical logic programming has the set  $\{\mathbf{f}, \mathbf{t}\}$  (false, true) as its intended truth space, and the usual semantics of a negation-free logic program is given by its unique minimal Herbrand model [10]. However, it is well known that the uniqueness and existence of such a model is no more guaranteed if we introduce nonmonotonic modes of negation.

Fitting [3] proposed to extend the classical  $\{\mathbf{f}, \mathbf{t}\}$  truth space to Kleene's three-valued logic [9], by introducing the logical value **unknown**, or  $\perp$  in our notation. In particular, he proposed to associate an operator  $\Phi_{\mathcal{P}}$  with any program  $\mathcal{P}$ , which is an extension to the Van Emden-Kowalski's immediate consequence operator  $T_{\mathcal{P}}$  [10] to the three-valued case.  $\Phi_{\mathcal{P}}$  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  $A$ . The nice consequence of this extension to three-valued logics is that the operator  $\Phi_{\mathcal{P}}$  is monotone with respect to the *knowledge ordering*, defined by  $\perp \preceq_k \mathbf{f}$  and  $\perp \preceq_k \mathbf{t}$ . It follows that  $\Phi_{\mathcal{P}}$  has a least fixed-point with respect to the knowledge ordering, which is a model of  $\mathcal{P}$ , and is known as the Kripke-Kleene semantics of  $\mathcal{P}$ .

But, the Kripke-Kleene semantics of logic programs with negation has been considered as to be too weak. A commonly accepted approach to provide stronger semantics consists in relying on the *Closed World Assumption* (CWA) to complete the available knowledge. Among the various approaches to the management of negation in logic programming, the *stable model* approach, introduced by Gelfond and Lifschitz [7] and later extended to the three-valued case by Przymusiński [11], has become one of the most widely studied and most commonly accepted approaches to negation in logic programming. Roughly, a set of ground atoms  $S$  is a stable model of a ground program  $\mathcal{P}$  if  $S = S'$ , where  $S'$  is computed according to the so-called Gelfond-Lifschitz transformation:

- (1) if an atom is head of no rule in  $\mathcal{P}$  then its truth value is false;
- (2) substitute (fix) in  $\mathcal{P}$  the negative literals by their evaluation with respect to  $S$ . Let  $\mathcal{P}'$  be the resulting *positive* program;
- (3) let  $S'$  be the minimal Herbrand model of  $\mathcal{P}'$ .

This approach defines a whole family of models. Remarkably, one among these stable models, indeed the *minimal* one according to the knowledge ordering, is considered as the favourite one and is one-to-one related with the so-called *well-founded semantics* [12, 13]. It is not unusual that, rather than to compute the whole set of stable models, one relies on the computation of the well-

founded semantics only.

It has been recognised that relying on a three-valued truth space is unnecessarily restrictive and that the whole can be nicely generalised to Belnap’s four-valued logic [2], denoted  $\mathcal{FOUR}$ , and more generally to the case where the underlying truth space is a *bilattice* [8]. Roughly, a bilattice is based on a set of truth values and two “orthogonal” orderings: the *truth ordering*  $\preceq_t$  and the *knowledge ordering*  $\preceq_k$ .  $\mathcal{FOUR}$ , is the simplest bilattices, where the truth space is  $\{\mathbf{f}, \mathbf{t}, \perp, \top\}$  ( $\top$  stands for *inconsistent*) and, additionally to the three-valued case,  $\mathbf{f} \preceq_t \top \preceq_t \mathbf{t}$  and both  $\mathbf{f} \preceq_k \top$  and  $\mathbf{t} \preceq_k \top$  hold. The two orderings are represented for Belnap’s logic in Figure 1.  $\mathcal{FOUR}$  will be used

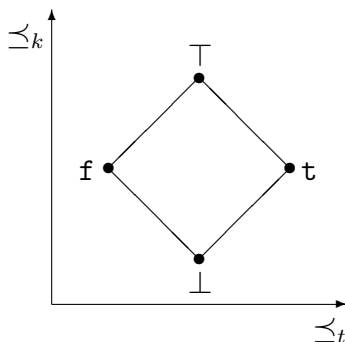


Fig. 1. The logic  $\mathcal{FOUR}$

to give a more intuitive comprehension of our approach to a reader who is not familiar with the notion of bilattice<sup>1</sup>.

In [4], Fitting generalises  $\Phi_{\mathcal{P}}$  by extending the Kripke-Kleene semantics to bilattices, while in [5, 6] he proposed a binary immediate consequence operator  $\Psi_{\mathcal{P}}$  as a generalisation of the Gelfond-Lifschitz transformation [7] to bilattices. Indeed, he extends the notions of stable models and well-founded semantics to the context of bilattices. The basic principle of  $\Psi_{\mathcal{P}}$ , as well as that of the Gelfond-Lifschitz transformation, is to separate the roles of positive and negative information, i.e.  $\Psi_{\mathcal{P}}$  accepts two input interpretations over a bilattice, the first one is used to assign meanings to positive literals, while the second one 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 anti-monotonic in the second argument (indeed, as the truth of a positive literal increases, the truth of its negation decreases). Computationally, he follows the idea of the Gelfond-Lifschitz transformation we have seen above: the idea is to fix an interpretation for negative information and to compute the least model of

<sup>1</sup> Arieli and Avron show in [1] 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 in [4] why  $\mathcal{FOUR}$  can be thought as the “home” of classical logic programming.

the resulting positive program with respect  $\preceq_t$ . To this end, Fitting [5] introduced the  $\Psi'_{\mathcal{P}}$  operator, which for a given interpretation  $I$  of negative literals, computes that model, i.e.  $\Psi'_{\mathcal{P}}(I) = \text{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathcal{P}}(x, I))$ . 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  $\mathcal{P}$ . Due to the above mentioned separation of positive and negative information, this approach avoids the kind of symmetry that is present in the Kripke-Kleene approach.

In this paper, we show that this separation is not necessary, at least for defining and computing the well-founded semantics. Indeed, we show that the well-founded semantics over bilattices can be defined in terms of  $\Phi_{\mathcal{P}}$  only. The main notion that we introduce is that of *support* with respect to a given interpretation  $I$ . The support of  $I$  determines in a principled way how much *false* knowledge, i.e. how much knowledge provided by the CWA, can “safely” be joined to  $I$  with respect to the program  $\mathcal{P}$ . More technically, starting from the *everywhere false interpretation*  $\mathbf{I}_{\mathbf{f}}$ , the support of  $I$  is the  $\preceq_k$  maximal false knowledge interpretation  $J$  such that  $J \preceq_k \mathbf{I}_{\mathbf{f}}$ , and  $J \preceq_k \Phi_{\mathcal{P}}(I \oplus J)$ , where  $\oplus$  denotes the join operator with respect to the knowledge ordering. So, if the current step of the computation is the interpretation  $I$ , we first compute the support  $J$  of  $I$  and, thus, determine how much false knowledge can compatibly be joined to  $I$ , and then activate the rules, by applying  $\Phi_{\mathcal{P}}$  with respect to the augmented knowledge represented by  $I \oplus J$ . We show that all fixed-points of  $\Psi'_{\mathcal{P}}$ , i.e. stable models of  $\mathcal{P}$ , are fixed-points of our new operator  $\Pi_{\mathcal{P}}$ , but that the reverse does not hold. Finally, we show that the least fixed-point of  $\Pi_{\mathcal{P}}$  with respect to the knowledge ordering coincides with the well-founded semantics of  $\mathcal{P}$ . Apart obtaining an alternative way to compute the well-founded semantics over bilattices, our approach does not require any separation between the roles of positive and negative information, but relies on a natural management of negation only. As our paper is rather technically oriented, we will not address the full impact of our new computation to the case where only classical logic programs and semantics are considered.

The remaining of the paper is organised as follows. In order to make the paper self-contained, in the next section, we will briefly define the notions of Belnap’s logic, bilattice, logic program, interpretation and Fitting’s immediate consequence operators  $\Phi_{\mathcal{P}}$ ,  $\Psi_{\mathcal{P}}$  and  $\Psi'_{\mathcal{P}}$ . In Section 3 we present our immediate consequence operator  $\Pi_{\mathcal{P}}$  and the main results. Section 4 concludes.

## 2 Preliminaries

**Bilattices.** We first recall the definition of bilattices and give some of their basic properties. A *bilattice* is a structure  $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$  where  $\mathcal{B}$  is a non-empty set and  $\preceq_t$  and  $\preceq_k$  are each partial orderings giving  $\mathcal{B}$  the structure of a

complete lattice with a top and bottom element. *Meet and join under  $\preceq_t$*  are denoted  $\wedge$  and  $\vee$ , and *meet and join under  $\preceq_k$*  are denoted  $\otimes$  and  $\oplus$ . *Top and bottom under  $\preceq_t$*  are denoted  $\mathbf{t}$  and  $\mathbf{f}$ , and the *top and the bottom under  $\preceq_k$*  are denoted  $\top$ ,  $\perp$ . We will assume that the bilattices are *distributive bilattices* in which all distributive laws connecting  $\wedge, \vee, \otimes$  and  $\oplus$  hold. We also assume that every bilattice satisfies the *interlacing conditions*, where each of the lattice operations  $\wedge, \vee, \otimes$  and  $\oplus$  is monotone w.r.t. both orderings. An example of interlacing condition is:  $x \preceq_t y$  and  $x' \preceq_t y'$  implies  $x \otimes x' \preceq_t y \otimes y'$ . Finally, we assume that each bilattice has a *negation*, which is mapping  $\neg$  that reverses the  $\preceq_t$  ordering, leaves unchanged the  $\preceq_k$  ordering, and  $\neg\neg x = x$ . Below, some properties that will be of use in this paper.

**Proposition 1**

- (1) If  $x \preceq_t y \preceq_t z$  then  $x \otimes z \preceq_k y$  and  $y \preceq_k x \oplus z$ ;
- (2) If  $x \preceq_k y \preceq_k z$  then  $x \wedge z \preceq_t y$  and  $y \preceq_t x \vee z$ ;
- (3) If  $x \preceq_t y$  then  $x \preceq_t x \otimes y \preceq_t y$  and  $x \preceq_t x \oplus y \preceq_t y$ ;
- (4) If  $x \preceq_k y$  and  $z \preceq_t y$  then  $z \oplus x \preceq_t y$ , thus if  $x \preceq_k y$  then  $\mathbf{f} \oplus x \preceq_t y$ ;
- (5) If  $x \preceq_k y$  and  $z \preceq_t x$  then  $z \otimes y \preceq_t x$ , thus if  $x \preceq_k y$  then  $\mathbf{f} \otimes y \preceq_t x$ .

**PROOF.** Point 1. and 2. are proven in [5]. Using the interlacing conditions: Point 3. is straightforward; for Point 4., we have  $x \oplus z \preceq_t x \oplus y = y$ ; for Point 5., we have  $z \otimes y \preceq_t x \otimes y = x$ .  $\square$

**Logic Programs and Kripke-Kleene semantics.** A logic program is defined as follows. A *formula* is an expression built up from the literals and the members of  $\mathcal{B}$  using  $\wedge, \vee, \otimes, \oplus, \exists$  and  $\forall$ . A *pure* formula is a formula that does not contain a member of  $\mathcal{B}$ . A *clause* is of the form  $P(x_1, \dots, x_n) \leftarrow \varphi(x_1, \dots, x_n)$ , where the pure atomic formula  $P(x_1, \dots, x_n)$  is the *head*, and the formula  $\varphi(x_1, \dots, x_n)$  is the *body*. It is assumed that the free variables of the body are among  $x_1, \dots, x_n$ . A *logic program*, always denoted with  $\mathcal{P}$ , is a finite set of clauses with no predicate letter appearing in the head of more than one clause. By  $\mathcal{P}^*$  we mean the set of all ground instances of members of  $\mathcal{P}$ , over the Herbrand base. A *classical* logic program is one whose underlying truth value space is the bilattice *FOUR* [2], and which does not involve  $\otimes, \oplus, \forall, \exists, \top$  and  $\perp$ .

Let  $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$  be a bilattice. By *interpretation* on the bilattice we mean a mapping  $I$  from ground atoms to members of  $\mathcal{B}$ : (i) for  $b \in \mathcal{B}$ ,  $I(b) = b$ ; (ii) for formulae  $\varphi$  and  $\varphi'$ ,  $I(\varphi \wedge \varphi') = I(\varphi) \wedge I(\varphi')$ , and similarly for  $\vee, \otimes, \oplus$  and  $\neg$ ; and (iii) we set  $I(\exists x \varphi(x)) = \vee \{I(\varphi(t)) : t \text{ closed term}\}$ , and similarly for universal quantification.

The family of all interpretations is denoted  $\mathcal{I}(\mathcal{B})$ . It is given two point wise

orderings: (i)  $I_1 \preceq_t I_2$  iff  $I_1(A) \preceq_t I_2(A)$ , for every ground atom  $A$ ; and (ii)  $I_1 \preceq_k I_2$  iff  $I_1(A) \preceq_k I_2(A)$ , for every ground atom  $A$ . An interpretation  $I$  is a *model* of  $\mathcal{P}$  provided, for each pure ground atom  $A$ , if  $A \leftarrow \varphi$  in  $\mathcal{P}^*$ ,  $I(A) = I(\varphi)$ , and otherwise  $I(A) = \mathbf{f}$ . Given two interpretations  $I, J$ , we define  $(I \wedge J)(\varphi) = I(\varphi) \wedge J(\varphi)$ , and similarly for the other operations. With  $\mathbf{I}_{\mathbf{f}}$  and  $\mathbf{I}_{\mathbf{t}}$  we will denote the bottom and top interpretation under  $\preceq_t$  (they map any atom into  $\mathbf{f}$  and  $\mathbf{t}$ , respectively). With  $\mathbf{I}_{\perp}$  and  $\mathbf{I}_{\top}$  we will denote the bottom and top interpretation under  $\preceq_k$  (they map any atom into  $\perp$  and  $\top$ , respectively). It is easy to see that the space of interpretations  $\langle \mathcal{I}(\mathcal{B}), \preceq_t, \preceq_k \rangle$  is a bilattice as well.

The mapping  $\Phi_{\mathcal{P}}: \mathcal{I}(\mathcal{B}) \rightarrow \mathcal{I}(\mathcal{B})$  is defined as follows. For  $I \in \mathcal{I}(\mathcal{B})$ ,  $\Phi_{\mathcal{P}}(I)$  is the interpretation such that

- (1) if the pure atom  $A$  is not the head of any member of  $\mathcal{P}^*$ ,  $\Phi_{\mathcal{P}}(I)(A) = \mathbf{f}$ ;
- (2) if  $A \leftarrow \varphi$  occurs in  $\mathcal{P}^*$ ,  $\Phi_{\mathcal{P}}(I)(A) = I(\varphi)$ .

Finally, a mapping  $f$  on a partial ordered space is *monotone* if  $x \preceq y$  implies  $f(x) \preceq f(y)$ ;  $f$  is *antitone* if  $x \preceq y$  implies  $f(y) \preceq f(x)$ .

**Theorem 2 ([5])** *The operator  $\Phi_{\mathcal{P}}$  is monotone under  $\preceq_k$ .*

As a consequence, from monotonicity in the  $\preceq_k$  ordering, since  $\mathcal{B}$  complete, from the Knaster-Tarski theorem it follows that  $\Phi_{\mathcal{P}}$  has least and greatest fixed-points under  $\preceq_k$ . Furthermore, it can easily be shown that an interpretation  $I$  is a model of a program  $\mathcal{P}$  iff  $I$  is a fixed-point of  $\Phi_{\mathcal{P}}$ .

**Fitting's immediate consequence operator  $\Psi'_{\mathcal{P}}$ .** Let  $I, J$  be interpretations in the bilattice  $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$ . The notion of *pseudo-interpretation*  $I \Delta J$  over the bilattice is defined as follows. For a pure ground atom  $A$ :

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

Pseudo-interpretations are extended to non-literals in the obvious way. The *extended immediate consequence operator*,  $\Psi_{\mathcal{P}}: \mathcal{I}(\mathcal{B}) \times \mathcal{I}(\mathcal{B}) \rightarrow \mathcal{I}(\mathcal{B})$ , is defined as follows. For  $I, J \in \mathcal{I}(\mathcal{B})$ ,  $\Psi_{\mathcal{P}}(I, J)$  is the interpretation, where

- (1) if the pure atom  $A$  is not the head of any member of  $\mathcal{P}^*$ ,  $\Psi_{\mathcal{P}}(I, J)(A) = \mathbf{f}$ ;
- (2) if  $A \leftarrow \varphi$  occurs in  $\mathcal{P}^*$ ,  $\Psi_{\mathcal{P}}(I, J)(A) = (I \Delta J)(\varphi)$ .

Note that by construction  $\Phi_{\mathcal{P}}(I) = \Psi_{\mathcal{P}}(I, I)$ .

**Theorem 3 ([5])** *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.

The *derived (or stability) operator* of  $\Psi_{\mathcal{P}}$  is the single input mapping  $\Psi'_{\mathcal{P}}$  given by:  $\Psi'_{\mathcal{P}}(I)$  is the smallest fixed-point, in the  $\preceq_t$  ordering, of the mapping  $\lambda x. \Psi_{\mathcal{P}}(x, I)$ , i.e.  $\Psi'_{\mathcal{P}}(I) = \text{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathcal{P}}(x, I))$ . By Theorem 3,  $\Psi'_{\mathcal{P}}$  is well defined and can be computed in the usual way:

**Theorem 4** *Let  $I$  be an interpretation. Consider the following sequence: for  $i \geq 0$ ,  $v_0^I = \mathbf{I}_{\mathbf{f}}$  and  $v_{i+1}^I = \Psi_{\mathcal{P}}(v_i^I, I)$ . Then  $v_i^I$  is monotone non decreasing under  $\preceq_t$  and converges to  $\Psi'_{\mathcal{P}}(I)$ .*

A *stable interpretation* for  $\mathcal{P}$  is a fixed-point of  $\Psi'_{\mathcal{P}}$ .

**Theorem 5 ([5])** *Every stable interpretation for  $\mathcal{P}$  is a model, called stable model of  $\mathcal{P}$ .*

It can be shown that the operator  $\Psi'_{\mathcal{P}}$  is monotone in the  $\preceq_k$  ordering, and antitone in the  $\preceq_t$  ordering. Since  $\mathcal{B}$  is complete, from the Knaster-Tarski theorem it follows that  $\Psi'_{\mathcal{P}}$  has a least and a greatest fixed-point under  $\preceq_k$ . The least and the greatest fixed-points of  $\Psi'_{\mathcal{P}}$  under  $\preceq_k$  are denoted by  $\mathbf{s}_{\mathcal{P}}^k$  and  $\mathbf{S}_{\mathcal{P}}^k$ , respectively.  $\mathbf{s}_{\mathcal{P}}^k$  is known as the *well-founded model* (wfm) of  $\mathcal{P}$ . The well-founded model of a program  $\mathcal{P}$  can be easily computed:

**Theorem 6** *Consider the following sequence: for  $n \geq 0$ ,  $W_0 = \mathbf{I}_{\perp}$  and  $W_{n+1} = \Psi'_{\mathcal{P}}(W_n)$ . Then  $W_n$  is monotone non decreasing under  $\preceq_k$  and converges to the least fixed-point,  $\mathbf{s}_{\mathcal{P}}^k$ , of  $\Psi'_{\mathcal{P}}$ .*

### 3 An alternative characterisation of the well-founded semantics

We have seen that the well-founded model of a program is the least fixed-point of  $\Psi'_{\mathcal{P}}$  under  $\preceq_k$  and, thus, is a fixed-point of  $\Phi_{\mathcal{P}}$ . In this section, we will give an alternative characterisation of that model. Our primary goal is to compute the well-founded model directly by relying on  $\Phi_{\mathcal{P}}$  rather than to go through  $\Psi'_{\mathcal{P}}$ . The main concept is the following.

**Definition 7 (safe)** *An interpretation  $J$  is safe w.r.t.  $\mathcal{P}$  and an interpretation  $I$  iff  $J \preceq_k \mathbf{I}_{\mathbf{f}}$  and  $J \preceq_k \Phi_{\mathcal{P}}(I \oplus J)$ .*

The idea is as following: let us consider an interpretation  $I$ , a program  $\mathcal{P}$  and consider the everywhere false interpretation  $\mathbf{I}_{\mathbf{f}}$ .  $\mathbf{I}_{\mathbf{f}}$  represents the CWA that asserts that every atom is supposed to be false by default. How much have we to weaken ( $J \preceq_k \mathbf{I}_{\mathbf{f}}$ ) the everywhere false knowledge  $\mathbf{I}_{\mathbf{f}}$  such that what remains ( $J$ ) is still enough to increase our knowledge w.r.t.  $I$  after one application of  $\Phi_{\mathcal{P}}$  ( $J \preceq_k \Phi_{\mathcal{P}}(I \oplus J)$ ), i.e. is compatible w.r.t.  $\mathcal{P}$  and  $I$ ? Let's try

to make the concept clearer with an example. Consider the following classical program  $\mathcal{P}$  over the Belnap's bilattice  $\mathcal{F}OUR$ ,  $\{A \leftarrow B \vee \mathbf{t}, B \leftarrow B\}$ , and an interpretation  $I$ , where  $I(A) = \mathbf{t}$  and  $I(B) = \perp$ . By definition we have  $\mathbf{I}_{\mathbf{f}}(A) = \mathbf{f}$  and  $\mathbf{I}_{\mathbf{f}}(B) = \mathbf{f}$ . Let us first consider the interpretation  $J = \mathbf{I}_{\mathbf{f}}$ . It follows that  $J$  is not safe. Indeed,  $J(B) = \mathbf{f} \preceq_k \Phi_{\mathcal{P}}(I \oplus J)(B) = \mathbf{f}$ , but  $J(A) = \mathbf{f} \not\preceq_k \Phi_{\mathcal{P}}(I \oplus J)(A) = \mathbf{t}$ . That is, we considered too much “default knowledge” about  $A$  which let us to an incompatibility w.r.t.  $\mathcal{P}$  and  $I$ . This means that we have to weakening our default knowledge about  $A$ , mapping  $A$  into  $\perp$ . Indeed, if we consider  $J'$  such that  $J'(A) = \perp$  and  $J'(B) = \mathbf{f}$ , then  $J'(B) = \mathbf{f} \preceq_k \Phi_{\mathcal{P}}(I \oplus J')(B) = \mathbf{f}$  and  $J'(A) = \perp \preceq_k \Phi_{\mathcal{P}}(I \oplus J')(A) = \mathbf{t}$ , i.e.  $J'$  is safe.

Among all possible safe interpretations w.r.t.  $\mathcal{P}$  and  $I$ , we will take the maximal under  $\preceq_k$ , which is unique as well.

**Definition 8 (support)** *The support, denoted  $s_{\mathcal{P}}(I)$ , w.r.t.  $\mathcal{P}$  and interpretation  $I$  is defined as  $s_{\mathcal{P}}(I) = \bigoplus\{J: J \text{ is safe w.r.t. } \mathcal{P} \text{ and } I\}$ .*

It is easy to show that the support is safe. In fact, given two safe interpretations  $J$  and  $J'$ , then  $J \oplus J' \preceq_k \mathbf{I}_{\mathbf{f}}$  and, from the monotonicity of  $\Phi_{\mathcal{P}}$  under  $\preceq_k$ ,  $J \oplus J' \preceq_k \Phi_{\mathcal{P}}(I \oplus J \oplus J')$  and, thus,  $J \oplus J'$  is safe. Therefore,  $s_{\mathcal{P}}(I)$  is safe.

As next, we address the issue of computing the support. We will define a specific function and show that a particular fixed-point of it is indeed the support. The *support function*, denoted  $\sigma_{\mathcal{P}}^I$ , w.r.t. a program  $\mathcal{P}$  and an interpretation  $I$  is the function mapping interpretations into interpretations defined as follows: for any interpretation  $J$ ,  $\sigma_{\mathcal{P}}^I(J) = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus J)$ . It is easy to verify that  $\sigma_{\mathcal{P}}^I$  is monotone w.r.t.  $\preceq_k$ . The following theorem specifies how to compute the support.

**Theorem 9** *Let  $I$  be an interpretation. Consider the iterated sequence of interpretations  $F_i^I$  defined as follows: for any  $i \geq 0$ ,  $F_0^I = \mathbf{I}_{\mathbf{f}}$  and  $F_{i+1}^I = \sigma_{\mathcal{P}}^I(F_i^I)$ . The sequence  $F_i^I$  ( $i$ ) is monotone non-increasing under  $\preceq_k$  and, thus, has a fixed-point  $F_{\omega}^I$ , for some limit ordinal  $\omega$ ; and ( $ii$ ) is monotone non-decreasing under  $\preceq_t$ . Furthermore, it follows that  $s_{\mathcal{P}}(I) = F_{\omega}^I$ .*

**PROOF.** Concerning ( $i$ ),  $F_1^I \preceq_k F_0^I$  and  $\sigma_{\mathcal{P}}^I$  is monotone under  $\preceq_k$ , so the sequence is non-increasing under  $\preceq_k$ . Therefore, the sequence has a fixed-point at the limit, say  $F_{\omega}^I$ . Concerning ( $ii$ ), from  $F_{i+1}^I \preceq_k F_i^I$ , by Proposition 1,  $F_i^I = F_i^I \otimes \mathbf{I}_{\mathbf{f}} \preceq_t F_{i+1}^I$ . Now, let us show that  $F_{\omega}^I$  is safe and  $\preceq_k$ -maximal.  $F_{\omega}^I = \sigma_{\mathcal{P}}^I(F_{\omega}^I) = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus F_{\omega}^I)$ . Therefore,  $F_{\omega}^I \preceq_k \mathbf{I}_{\mathbf{f}}$  and  $F_{\omega}^I \preceq_k \Phi_{\mathcal{P}}(I \oplus F_{\omega}^I)$ , so  $F_{\omega}^I$  is safe w.r.t.  $\mathcal{P}$  and  $I$ . Consider any  $X$  safe w.r.t.  $\mathcal{P}$  and  $I$ . We show by induction on  $i$  that  $X \preceq_k F_i^I$  and, thus, at the limit  $X \preceq_k F_{\omega}^I$ , so  $F_{\omega}^I$  is  $\preceq_k$ -maximal. ( $i$ ) Case  $i = 0$ . By definition,  $X \preceq_k \mathbf{I}_{\mathbf{f}} = F_0^I$ . ( $ii$ ) Induction step:

suppose  $X \preceq_k F_i^I$ . Since  $X$  is safe, we have  $X \preceq_k X \otimes X \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus X)$ . By induction,  $X \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus F_i^I) = F_{i+1}^I$ .  $\square$

**Theorem 10**  $s_{\mathcal{P}}$  is monotone under  $\preceq_k$ .

**PROOF.** Consider two interpretations  $I$  and  $J$ , where  $I \preceq_k J$ . Consider the two sequences  $F_i^I$  and  $F_i^J$  of the computation of the support w.r.t.  $I$  and  $J$ , respectively, according to Theorem 9. We show now by induction on  $i$  that  $F_i^I \preceq_k F_i^J$  and, thus, at the limit  $s_{\mathcal{P}}(I) \preceq_k s_{\mathcal{P}}(J)$ . (i) Case  $i = 0$ . By definition,  $F_0^I = \mathbf{I}_{\mathbf{f}} \preceq_k \mathbf{I}_{\mathbf{f}} = F_0^J$ . (ii) Induction step: suppose  $F_i^I \preceq_k F_i^J$ . By monotonicity under  $\preceq_k$  of  $\Phi_{\mathcal{P}}$  and the induction hypothesis,  $F_{i+1}^I = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus F_i^I) \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(J \oplus F_i^J) = F_{i+1}^J$ , which concludes.  $\square$

We are now ready to specify the immediate consequence operator whose least fixed-point under  $\preceq_k$  is the well-founded model of a program. Applying that operator on an interpretation  $I$  consists in completing (joining)  $I$  with the “maximal safe part of the CWA” w.r.t.  $\mathcal{P}$  and  $I$ , and then activating the rules of  $\mathcal{P}$  on that joined interpretation.

**Definition 11 (immediate consequence operator)** *The immediate consequence operator w.r.t.  $\mathcal{P}$  maps interpretations into interpretations and is defined as  $\Pi_{\mathcal{P}}(I) = \Phi_{\mathcal{P}}(I \oplus s_{\mathcal{P}}(I))$ .*

It is easy to verify that  $\Pi_{\mathcal{P}}$  is monotone w.r.t.  $\preceq_k$ , as the involved operators are monotone. Therefore, from the Knaster-Tarski theorem,  $\Pi_{\mathcal{P}}$  has a least (and a greatest) fixed-point under  $\preceq_k$ . The least fixed-point of  $\Pi_{\mathcal{P}}$  under  $\preceq_k$  is denoted by  $\pi_{\mathcal{P}}^k$ . Therefore,

**Theorem 12 (immediate consequence iteration)** *Consider the immediate consequence iteration w.r.t.  $\mathcal{P}$ , i.e. the sequence of interpretations  $I_n$  defined as follows: for any  $n \geq 0$ ,  $I_0 = \mathbf{I}_{\perp}$  and  $I_{n+1} = \Pi_{\mathcal{P}}(I_n)$ . It follows that  $I_n$  is monotone non-decreasing under  $\preceq_k$  and converges to the least fixed-point  $\pi_{\mathcal{P}}^k$  of  $\Pi_{\mathcal{P}}$ .*

**Note 1** *In the rest of this paper,*

- with  $I_n$  we will indicate the  $n$ -th iteration of the immediate consequence iteration w.r.t.  $\mathcal{P}$ , according to Theorem 12;
- with  $F_i^I$  we will indicate the  $i$ -th iteration of the computation of the support w.r.t.  $I$ , according to Theorem 9;
- with  $v_i^I$  we will indicate the  $i$ -th iteration of the computation of  $\Psi'_{\mathcal{P}}(I)$ , according to Theorem 4

In the following we will show that fixed-points of  $\Pi_{\mathcal{P}}$  are models of  $\mathcal{P}$ .

**Lemma 13**

- (1) For any  $n$ ,  $s_{\mathcal{P}}(I_n) = \mathbf{I}_{\mathbf{f}} \otimes I_{n+1} \preceq_k I_{n+1}$ ;
- (2) If  $I = \Pi_{\mathcal{P}}(I)$  then  $s_{\mathcal{P}}(I) = \mathbf{I}_{\mathbf{f}} \otimes I \preceq_k I$ ;
- (3) For any  $n$ ,  $s_{\mathcal{P}}(I_n) \preceq_t I_n$  and  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_t I_n$ .

**PROOF.** By construction,  $s_{\mathcal{P}}(I_n) = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n)) = \mathbf{I}_{\mathbf{f}} \otimes I_{n+1} \preceq_k I_{n+1}$ . Similarly,  $s_{\mathcal{P}}(I) = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I \oplus s_{\mathcal{P}}(I)) = \mathbf{I}_{\mathbf{f}} \otimes \Pi_{\mathcal{P}}(I) = \mathbf{I}_{\mathbf{f}} \otimes I \preceq_k I$ . Finally, from  $I_n \preceq_k I_{n+1}$  and from Proposition 1,  $s_{\mathcal{P}}(I_n) = I_{n+1} \otimes \mathbf{I}_{\mathbf{f}} \preceq_t I_n$  and, thus,  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_t I_n$ .  $\square$

**Theorem 14** Every fixed-point  $I$  of  $\Pi_{\mathcal{P}}$  is a model of  $\mathcal{P}$ .

**PROOF.** By definition and by Lemma 13,  $I = \Pi_{\mathcal{P}}(I) = \Phi_{\mathcal{P}}(I \oplus s_{\mathcal{P}}(I)) = \Phi_{\mathcal{P}}(I)$ . Therefore,  $I$  is a fixed-point of  $\Phi_{\mathcal{P}}$  and, thus, is a model of  $\mathcal{P}$ .  $\square$

We are now ready to present the main results of our work:

- every stable model is a fixed-point of  $\Pi_{\mathcal{P}}$ , but not vice-versa;
- the well-founded model  $\mathbf{s}_{\mathcal{P}}^k$ , i.e. the least fixed-point under  $\preceq_k$  of  $\Psi'_{\mathcal{P}}$ , coincides with the least fixed-point under  $\preceq_k$  of  $\Pi_{\mathcal{P}}$ .

At first, let us present some auxiliary results.

**Lemma 15**

- (1) If  $I \preceq_t J$  and  $J \preceq_k I$ , then for any  $x$ ,  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, I) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, J)$ ;
- (2) If  $J \preceq_t I$  and  $J \preceq_k I$ , then for any  $x$ ,  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I, x) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(J, x)$ .

**PROOF.** For the first point, using the antimonotonicity of  $\Psi_{\mathcal{P}}$  w.r.t.  $\preceq_t$  for its second argument, we have  $\mathbf{I}_{\mathbf{f}} \preceq_t \Psi_{\mathcal{P}}(x, J) \preceq_t \Psi_{\mathcal{P}}(x, I)$ . From Proposition 1, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, I) \preceq_k \Psi_{\mathcal{P}}(x, J)$ . Using the interlacing conditions, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, I) \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, J)$ . Now, using the monotonicity of  $\Psi_{\mathcal{P}}$  w.r.t.  $\preceq_k$  and the interlacing conditions, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, J) \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, I)$ . Therefore,  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, I) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(x, J)$ .

Similarly, for the second point, using the monotonicity of  $\Psi_{\mathcal{P}}$  w.r.t.  $\preceq_t$  for its first argument, we have  $\mathbf{I}_{\mathbf{f}} \preceq_t \Psi_{\mathcal{P}}(J, x) \preceq_t \Psi_{\mathcal{P}}(I, x)$ . From Proposition 1, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I, x) \preceq_k \Psi_{\mathcal{P}}(J, x)$ . Using the interlacing conditions, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I, x) \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(J, x)$ . Now, using the monotonicity of  $\Psi_{\mathcal{P}}$  w.r.t.  $\preceq_k$  and the interlacing conditions, we have  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(J, x) \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I, x)$ . Therefore,  $\mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I, x) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(J, x)$ .  $\square$

Now, let us prove that every stable model is a fixed-point of  $\Pi_{\mathcal{P}}$ .

**Lemma 16** *If  $I = \Phi_{\mathcal{P}}(I)$  then  $F_i^I \preceq_t I$ , for all  $i$ .*

**PROOF.** We show by induction on  $i$  that  $F_i^I \preceq_t I$ . (i) Case  $i = 0$ .  $F_0^I = \mathbf{I}_{\mathbf{f}} \preceq_t I$ . (ii) Induction step: let us assume that  $F_i^I \preceq_t I$  holds. From Proposition 1,  $F_i^I \preceq_t F_i^I \oplus I \preceq_t I$  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 15 that  $F_{i+1}^I = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I \oplus I, F_i^I \oplus I) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I, I)$ . By induction  $F_i^I \preceq_t I$ , so from  $I = \Phi_{\mathcal{P}}(I)$ ,  $F_{i+1}^I = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I, I) \preceq_t \Psi_{\mathcal{P}}(F_i^I, I) \preceq_t \Psi_{\mathcal{P}}(I, I) = \Phi_{\mathcal{P}}(I) = I$  follows.  $\square$

**Lemma 17** *If  $I = \Psi'_{\mathcal{P}}(I)$  then  $s_{\mathcal{P}}(I) \preceq_k I$ .*

**PROOF.** We show by induction on  $i$ , that  $F_i^I \preceq_k v_i^I$ . Therefore, at the limit,  $s_{\mathcal{P}}(I) \preceq_k \Psi'_{\mathcal{P}}(I) = I$ . (i) Case  $i = 0$ .  $F_0^I = \mathbf{I}_{\mathbf{f}} \preceq_k \mathbf{I}_{\mathbf{f}} = v_0^I$ . (ii) Induction step: let us assume that  $F_i^I \preceq_k v_i^I$  holds. By Theorem 5,  $I = \Psi'_{\mathcal{P}}(I)$  implies  $I = \Phi_{\mathcal{P}}(I)$ . So, from Lemma 16, we have  $F_i^I \preceq_t I$  and, thus, by Proposition 1,  $F_i^I \preceq_t F_i^I \oplus I \preceq_t I$  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 15,  $F_{i+1}^I = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I \oplus I, F_i^I \oplus I) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I, I)$ . By induction  $F_i^I \preceq_t v_i^I$ , so we have  $F_{i+1}^I = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^I, I) \preceq_k \Psi_{\mathcal{P}}(F_i^I, I) \preceq_k \Psi_{\mathcal{P}}(v_i^I, I) = v_{i+1}^I$ , which concludes.  $\square$

**Theorem 18** *Every stable model is fixed-point of  $\Pi_{\mathcal{P}}$ .*

**PROOF.** From Lemma 17 and from Theorem 5, we have  $\Pi_{\mathcal{P}}(I) = \Phi_{\mathcal{P}}(I \oplus s_{\mathcal{P}}(I)) = \Phi_{\mathcal{P}}(I) = I$ .  $\square$

As a consequence,  $\mathbf{s}_{\mathcal{P}}^k$  is a fixed-point of  $\Pi_{\mathcal{P}}$ . Therefore,

**Corollary 19**  $\pi_{\mathcal{P}}^k \preceq_k \mathbf{s}_{\mathcal{P}}^k$ .

Now, let us prove the second result, i.e. that  $\pi_{\mathcal{P}}^k = \mathbf{s}_{\mathcal{P}}^k$ . By Corollary 19, it remains to show that  $\mathbf{s}_{\mathcal{P}}^k \preceq_k \pi_{\mathcal{P}}^k$ . The outline of the proof of the latter property is as follows. We will first show that  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_t \pi_{\mathcal{P}}^k$  and  $\pi_{\mathcal{P}}^k \preceq_t \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ . Therefore,  $\pi_{\mathcal{P}}^k$  is a fixed-point of  $\Psi'_{\mathcal{P}}$ , i.e.  $\pi_{\mathcal{P}}^k = \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ . But,  $\mathbf{s}_{\mathcal{P}}^k$  is the least fixed-point of  $\Psi'_{\mathcal{P}}$  under  $\preceq_k$  and, thus,  $\mathbf{s}_{\mathcal{P}}^k \preceq_k \pi_{\mathcal{P}}^k$  holds.

**Lemma 20**  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_t \pi_{\mathcal{P}}^k$ .

**PROOF.** By definition and Lemma 13,  $\pi_{\mathcal{P}}^k = \Pi_{\mathcal{P}}(\pi_{\mathcal{P}}^k) = \Phi_{\mathcal{P}}(\pi_{\mathcal{P}}^k \oplus s_{\mathcal{P}}(\pi_{\mathcal{P}}^k)) = \Phi_{\mathcal{P}}(\pi_{\mathcal{P}}^k) = \Psi_{\mathcal{P}}(\pi_{\mathcal{P}}^k, \pi_{\mathcal{P}}^k)$ , i.e.  $\pi_{\mathcal{P}}^k$  is a fixed-point of  $\Psi_{\mathcal{P}}$ . But  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$  is the

least fixed-point under  $\preceq_t$  of  $\lambda x. \Psi_{\mathcal{P}}(x, \pi_{\mathcal{P}}^k)$ , i.e.  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k) = \text{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathcal{P}}(x, \pi_{\mathcal{P}}^k))$ . Therefore,  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_t \pi_{\mathcal{P}}^k$  follows.  $\square$

**Lemma 21** *For any  $n$  and  $i$ ,  $F_i^{I_n} \preceq_k v_i^{I_n}$  holds. Therefore, at the limit  $s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n)$  and, thus,  $s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_k \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$  holds as well.*

**PROOF.** We know that  $v_i^{I_n}$  converges to  $\Psi'_{\mathcal{P}}(I_n)$  (Theorem 4). We show by induction on  $i$  that  $F_i^{I_n} \preceq_k v_i^{I_n}$ . Therefore, at the limit  $s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n)$  and, thus,  $s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_k \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$  holds as well. (i) Case  $i = 0$ .  $F_0^{I_n} = \mathbf{I}_{\mathbf{f}} \preceq_k \mathbf{I}_{\mathbf{f}} = v_0^{I_n}$ . (ii) Induction step: assume that  $F_i^{I_n} \preceq_k v_i^{I_n}$ . By definition,  $F_{i+1}^{I_n} = \mathbf{I}_{\mathbf{f}} \otimes \Phi_{\mathcal{P}}(I_n \oplus F_i^{I_n}) = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(I_n \oplus F_i^{I_n}, I_n \oplus F_i^{I_n})$ . We have  $F_i^{I_n} \preceq_k I_n \oplus F_i^{I_n}$ . By Theorem 9 and Lemma 13,  $F_i^{I_n} \preceq_t I_n$  and, thus,  $F_i^{I_n} \preceq_t I_n \oplus F_i^{I_n}$ . Similarly,  $I_n \preceq_k I_n \oplus F_i^{I_n}$  and  $I_n \oplus F_i^{I_n} \preceq_t I_n$ , so by Lemma 15,  $F_{i+1}^{I_n} = \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(F_i^{I_n}, I_n)$  follows. By the induction hypothesis we know that  $F_i^{I_n} \preceq_k v_i^{I_n}$ . Therefore,  $F_{i+1}^{I_n} \preceq_k \mathbf{I}_{\mathbf{f}} \otimes \Psi_{\mathcal{P}}(v_i^{I_n}, I_n) \preceq_k \Psi_{\mathcal{P}}(v_i^{I_n}, I_n) = v_{i+1}^{I_n}$  follows, which concludes.  $\square$

**Lemma 22** *For any  $n$ ,  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  and, thus, at the limit  $\pi_{\mathcal{P}}^k \preceq_k \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ .*

**PROOF.** We show by induction on  $n$  that  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . (i) Case  $n = 0$ .  $I_0 \oplus s_{\mathcal{P}}(I_0) = s_{\mathcal{P}}(I_0) = s_{\mathcal{P}}(\mathbf{I}_{\perp})$ . By Lemma 21 and the monotonicity of  $\Psi'_{\mathcal{P}}$  under  $\preceq_k$ ,  $s_{\mathcal{P}}(I_0) \preceq_k \Psi'_{\mathcal{P}}(\mathbf{I}_{\perp}) \preceq_k \Psi'_{\mathcal{P}}(s_{\mathcal{P}}(I_0))$ . (ii) Induction step: assume that  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . By definition,  $I_{n+1} = \Phi_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n)) = \Psi_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n), I_n \oplus s_{\mathcal{P}}(I_n))$ . By induction,  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ , therefore  $I_{n+1} \preceq_k \Psi_{\mathcal{P}}(\Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n)), I_n \oplus s_{\mathcal{P}}(I_n)) = \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . But,  $I_n \preceq_k I_{n+1}$  and by Theorem 10  $s_{\mathcal{P}}(I_n) \preceq_k s_{\mathcal{P}}(I_{n+1})$  and, thus  $I_{n+1} \preceq_k \Psi'_{\mathcal{P}}(I_{n+1} \oplus s_{\mathcal{P}}(I_{n+1}))$ . Therefore, from Lemma 21,  $I_{n+1} \oplus s_{\mathcal{P}}(I_{n+1}) \preceq_k \Psi'_{\mathcal{P}}(I_{n+1} \oplus s_{\mathcal{P}}(I_{n+1})) \oplus s_{\mathcal{P}}(I_{n+1}) = \Psi'_{\mathcal{P}}(I_{n+1} \oplus s_{\mathcal{P}}(I_{n+1}))$ , which concludes.

Furthermore, by Lemma 13,  $s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_k \pi_{\mathcal{P}}^k$  and, thus,  $\pi_{\mathcal{P}}^k = \pi_{\mathcal{P}}^k \oplus s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_k \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k \oplus s_{\mathcal{P}}(\pi_{\mathcal{P}}^k)) = \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ .  $\square$

Now we are ready to prove that  $\pi_{\mathcal{P}}^k \preceq_t \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$  holds.

**Lemma 23** *For any  $n$  and  $i$ ,  $I_n \oplus F_i^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  holds. Therefore, at the limit  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  and, thus,  $\pi_{\mathcal{P}}^k \preceq_t \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ .*

**PROOF.** We show by induction on  $i$  that  $I_n \oplus F_i^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . (i) Case  $i = 0$ . Let us show that  $I_n \oplus F_0^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  holds. From Lemma 22,  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . From Proposition 1,  $I_n \oplus s_{\mathcal{P}}(I_n) \oplus \mathbf{I}_{\mathbf{f}} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . Since  $s_{\mathcal{P}}(I_n) \preceq_k \mathbf{I}_{\mathbf{f}}$ ,  $I_n \oplus F_0^{I_n} = I_n \oplus \mathbf{I}_{\mathbf{f}} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  follows. (ii) Induction step: assume  $I_n \oplus F_i^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ .

$F_{i+1}^{I_n} = \mathbf{I}_f \otimes \Phi_{\mathcal{P}}(I_n \oplus F_i^{I_n}) = \mathbf{I}_f \otimes \Psi_{\mathcal{P}}(I_n \oplus F_i^{I_n}, I_n \oplus F_i^{I_n})$ . From Theorem 9,  $I_n \oplus F_i^{I_n} \preceq_t I_n \oplus s_{\mathcal{P}}(I_n)$  follows. As  $I_n \oplus s_{\mathcal{P}}(I_n) \preceq_k I_n \oplus F_i^{I_n}$ , by Lemma 15,  $F_{i+1}^{I_n} = \mathbf{I}_f \otimes \Psi_{\mathcal{P}}(I_n \oplus F_i^{I_n}, I_n \oplus s_{\mathcal{P}}(I_n))$  follows. By induction,  $I_n \oplus F_i^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  and, thus,  $F_{i+1}^{I_n} \preceq_t \mathbf{I}_f \otimes \Psi_{\mathcal{P}}(\Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n)), I_n \oplus s_{\mathcal{P}}(I_n)) = \mathbf{I}_f \otimes \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n)) \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . It follows that  $I_n \oplus F_{i+1}^{I_n} \preceq_t I_n \oplus \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ . Finally, from Lemma 22,  $I_n \preceq_k \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$  and, thus,  $I_n \oplus F_{i+1}^{I_n} \preceq_t \Psi'_{\mathcal{P}}(I_n \oplus s_{\mathcal{P}}(I_n))$ , which concludes.

Furthermore, by Lemma 13,  $s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_k \pi_{\mathcal{P}}^k$  and, thus,  $\pi_{\mathcal{P}}^k = \pi_{\mathcal{P}}^k \oplus s_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_t \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k \oplus s_{\mathcal{P}}(\pi_{\mathcal{P}}^k)) = \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ .  $\square$

**Lemma 24**  $\pi_{\mathcal{P}}^k = \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$  and, thus,  $\mathbf{s}_{\mathcal{P}}^k \preceq_k \pi_{\mathcal{P}}^k$  holds.

**PROOF.** By Lemma 20,  $\Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k) \preceq_t \pi_{\mathcal{P}}^k$ . By Lemma 23,  $\pi_{\mathcal{P}}^k \preceq_t \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ . Therefore,  $\pi_{\mathcal{P}}^k = \Psi'_{\mathcal{P}}(\pi_{\mathcal{P}}^k)$ , i.e.  $\pi_{\mathcal{P}}^k$  is a fixed-point of  $\Psi'_{\mathcal{P}}$ . Since  $\mathbf{s}_{\mathcal{P}}^k$  is the least fixed-point of  $\Psi'_{\mathcal{P}}$  w.r.t.  $\preceq_k$ , it follows that  $\mathbf{s}_{\mathcal{P}}^k \preceq_k \pi_{\mathcal{P}}^k$ .  $\square$

We conclude with

**Theorem 25**  $\pi_{\mathcal{P}}^k = \mathbf{s}_{\mathcal{P}}^k$

**PROOF.** By Corollary 19,  $\pi_{\mathcal{P}}^k \preceq_k \mathbf{s}_{\mathcal{P}}^k$ . By Lemma 24,  $\mathbf{s}_{\mathcal{P}}^k \preceq_k \pi_{\mathcal{P}}^k$ . Therefore,  $\pi_{\mathcal{P}}^k = \mathbf{s}_{\mathcal{P}}^k$ .  $\square$

Below, some examples to illustrate the results.

**Example 26** Using the bilattice *FOUR*, consider the following program  $\mathcal{P}$ .

$$\begin{aligned} A &\leftarrow A \vee \neg B \\ B &\leftarrow \neg C \\ C &\leftarrow C \end{aligned}$$

The well-founded model of  $\mathcal{P}$  is  $\mathbf{s}_{\mathcal{P}}^k(A) = \mathbf{f}$ ,  $\mathbf{s}_{\mathcal{P}}^k(B) = \mathbf{t}$ ,  $\mathbf{s}_{\mathcal{P}}^k(C) = \mathbf{f}$ . Let us verify that indeed  $\mathbf{s}_{\mathcal{P}}^k$  is computed as the least fixed-point of  $\Psi'_{\mathcal{P}}$  (according to Theorem 6) and that  $\mathbf{s}_{\mathcal{P}}^k = \pi_{\mathcal{P}}^k$  (according Theorem 25 and Definition 11). The  $\Psi'_{\mathcal{P}}$  computation is shown in Table 1, where  $W_4 = W_3 = \mathbf{s}_{\mathcal{P}}^k$ . In it, the columns in the right part illustrate the progress of the  $W_i$ 's, as specified in Theorem 6, whereas the columns in the left part illustrate the computation of each  $\Psi'_{\mathcal{P}}(W_i)$ , as specified in Theorem 4. On the other hand, Table 1 also shows the  $\Pi_{\mathcal{P}}$  computation of  $\pi_{\mathcal{P}}^k$ . As expected,  $I_2 = I_3 = \pi_{\mathcal{P}}^k = \mathbf{s}_{\mathcal{P}}^k$ : the columns in the

Table 1.

The  $\Psi'_{\mathcal{P}}$  computation of  $s_{\mathcal{P}}^k$ . The  $\Pi_{\mathcal{P}}$  computation of  $\pi_{\mathcal{P}}^k$ .

$v_i^{W_j}$	A	B	C	A	B	C	$W_j$
$v_0^{W_0}$	f	f	f	⊥	⊥	⊥	$W_0$
$v_1^{W_0}$	⊥	⊥	f				
$v_2^{W_0}$	⊥	⊥	f				
$v_0^{W_1}$	f	f	f	⊥	⊥	f	$W_1$
$v_1^{W_1}$	⊥	t	f				
$v_2^{W_1}$	⊥	t	f				
$v_0^{W_2}$	f	f	f	⊥	t	f	$W_2$
$v_1^{W_2}$	f	t	f				
$v_2^{W_2}$	f	t	f				
$v_0^{W_3}$	f	f	f	f	t	f	$W_3$
$v_1^{W_3}$	f	t	f				
$v_2^{W_3}$	f	t	f				
				f	t	f	$W_4$

$F_i^{I_n}$	A	B	C	A	B	C	$I_n$
$F_0^{I_0}$	f	f	f	⊥	⊥	⊥	$I_0$
$F_1^{I_0}$	⊥	⊥	f				
$F_2^{I_0} = s_{\mathcal{P}}(I_0)$	⊥	⊥	f				
$F_0^{I_1}$	f	f	f	⊥	t	f	$I_1$
$F_1^{I_1}$	f	⊥	f				
$F_2^{I_1} = s_{\mathcal{P}}(I_1)$	f	⊥	f				
$F_0^{I_2}$	f	f	f	f	t	f	$I_2$
$F_1^{I_2}$	f	⊥	f				
$F_2^{I_2} = s_{\mathcal{P}}(I_2)$	f	⊥	f				
				f	t	f	$I_3$

right part illustrate the progress of  $I_n$ , as specified in Theorem 12, whereas the columns in the left part illustrate the computation of the support, as specified in Theorem 9. Note how the  $\pi_{\mathcal{P}}^k$  computations converges more rapidly.

**Example 27** Consider  $\mathcal{P} = \{A \leftarrow A\}$ . Consider the interpretations  $I_{\top}, I_{\perp}$  and  $I_{\mathbf{f}}$ . It can easily be shown that the following hold:

- (1)  $I_{\top} = \Pi_{\mathcal{P}}(I_{\top})$ , but  $I_{\top} \neq \Psi'_{\mathcal{P}}(I_{\top}) = I_{\mathbf{f}}$ ; and
- (2)  $I_{\perp} = \Phi_{\mathcal{P}}(I_{\perp})$ , but  $I_{\perp} \neq \Pi_{\mathcal{P}}(I_{\perp})$ .

That is, there are fixed-points of  $\Pi_{\mathcal{P}}$ , which are not fixed-points of  $\Psi'_{\mathcal{P}}$  and there are models of  $\mathcal{P}$ , which are not fixed-points of  $\Pi_{\mathcal{P}}$ .  $\square$

## 4 Conclusions

In this paper we have shown that the well-founded semantics of logic programs evaluated over bilattices can be defined in terms of an extension of the Kripke-Kleene semantics, i.e. by an extension of the  $\Phi_{\mathcal{P}}$  operator. As a consequence, the well-known and long studied separation of positive and negative information in the Gelfond-Lifschitz transformation is no longer necessary and we can revert to the usual interpretation of negation. Our main concept is the notion of support with respect to a given interpretation  $I$ , which determines in a principled way how much *false* knowledge can “safely” be joined to  $I$  with respect to the program  $\mathcal{P}$ .

We envisage two major topics for future research. Firstly, we did not exploit the impact of the new characterisation to the special case where the underlying truth space is Belnap’s *FOUR*, and logic programs are classical ones, neither from a properties point of view, neither from an computational point of view,

i.e. we did not investigate whether our algorithm results in a new algorithm for computing the well-founded semantics for classical logic programs or reverts to existing ones. Secondly, we were not able yet to determine whether we can extend  $\Phi_{\mathcal{P}}$  such that to determine all stable models, even if there is strong evidence that the adopted approach may be generalised. It is not clear yet whether either the definition of support, the definition of  $\Pi_{\mathcal{P}}$  or both have to be changed in order to have a one-to-one relationship between stable models and fixed-points of  $\Pi_{\mathcal{P}}$ . This, ultimately, would give us the possibility to view the stable models semantics from a different perspective.

## References

- [1] O. Arieli and A. Avron. The value of the four values. *Artificial Intelligence Journal*, 102(1):97–141, 1998.
- [2] N. D. Belnap. A useful four-valued logic. In G. Epstein and J. M. Dunn, editors, *Modern uses of multiple-valued logic*, pages 5–37. Reidel, Dordrecht, NL, 1977.
- [3] M .C. Fitting. A Kripke-Kleene-semantics for general logic programs. *Journal of Logic Programming*, 2:295–312, 1985.
- [4] M. C. Fitting. Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11:91–116, 1991.
- [5] M. C. Fitting. The family of stable models. *Journal of Logic Programming*, 17:197–225, 1993.
- [6] M. C. Fitting. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science*, 21(3):25–51, 2002.
- [7] M. Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In R. A. Kowalski and K. Bowen, editors, *Proceedings of the Fifth International Conference on Logic Programming*, pages 1070–1080, Cambridge, Massachusetts, 1988. The MIT Press.
- [8] M. L. Ginsberg. Multi-valued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
- [9] S. C. Kleene. *Introduction to Meta-Mathematics*. Van Nostran, Princeton, 1952.
- [10] J. W. Lloyd. *Foundations of Logic Programming*. Springer, Heidelberg, RG, 1987.
- [11] T. C. Przymusiński. Extended stable semantics for normal and disjunctive programs. In D. H. D. Warren and P. Szeredi, editors, *Proceedings of the Seventh International Conference on Logic Programming*, pages 459–477. MIT Press, 1990.
- [12] T. C. Przymusiński. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.
- [13] A. van Gelder, K. A. Ross, and J. S. Schlimpf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, January 1991.