Representing Imperfect Information through Extended Logic Programs in Multivalued Logics

Daniel Stamate
Department of Computing, Goldsmiths College
University of London, UK
E-mail d.stamate@doc.gold.ac.uk

Abstract

The basis of (default) negative information in the well-founded semantics is given by the so-called unfounded sets, used to complete missing information from a program through a kind of pessimistic assumption. We extend this concept by considering optimistic, pessimistic, skeptical and inconsistent assumptions in the context of multivalued logics given by bilattices. The extended well-founded semantics we define for general logic programs in bilattices is capable to express imperfect information considered to be missing/incomplete, uncertain and/or inconsistent. We provide a method of computing the semantics and show that, for different assumptions, it captures the Kripke-Kleene semantics, the well-founded semantics and Fitting's least multivalued stable model. We show also that the complexity of the computation of our semantics is polynomial time in a useful class of so-called limited bilattices.

Keywords: logic programs, imperfect information, multivalued logics, bilattices, assumptions

1 Introduction

One of the most used assumptions in logic programming and deductive databases is the so-called Closed World Assumption (CWA), according to which the atoms that cannot be inferred with the rules are considered to be false (i.e. a pessimistic assumption). Such assumptions are needed as the conventional logic programs with negation can be seen as incomplete logic theories, i.e. we cannot always infer any ground atom A or its negation from a logic program. In order to enrich such a theory we can make assumptions on the logical values of atoms that cannot be inferred from the rules. This is similar to the process of reasoning by default.

One of the most successful semantics of conventional logic programs based on the CWA is the well-founded semantics [19]. However, the CWA is not applicable in all circumstances when information is handled, as for example in a legal case, where a person should be considered innocent unless the contrary is proved (i.e. an optimistic assumption). That is, all the semantics based on the CWA, in particular the well-founded semantics, would behave inadequately in such a case.

In this paper we extend the well-founded semantics definition in order for it to be based also on alternative assumptions, in particular on an optimistic assumption, according to which, if in doubt then assume true.

Let us illustrate this using the following legal case example represented through the set of rules and facts P :

$$\begin{array}{lcl} \text{charge(X)} & \leftarrow & \text{suspect(X)} \wedge \neg\text{innocent(X)} \\ \text{free(X)} & \leftarrow & \text{suspect(X)} \wedge \text{innocent(X)} \\ \text{innocent(X)} & \leftarrow & \exists \text{Y(alibi(X,Y)} \wedge \neg\text{relatives(X,Y))} \\ \text{suspect(John)} & \leftarrow & \textit{true} \end{array}$$

The only assertion made in the program is that John is suspect, but we know nothing as to whether he is innocent.

If we consider the pessimistic assumption, then we are led to assume that John is not innocent, and we can infer that John must not be freed, and must be charged. If, on the other hand, we consider the skeptical assumption, i.e. we assume nothing about the innocence of John, then we can infer nothing as to whether he must be freed or charged.

If we consider the optimistic assumption then *innocent(John)* is true and we can infer that John must be freed, and must not be charged.

A fourth approach, less intuitive than the previous ones, is that in which in doubt we assume the value inconsistent (i.e. an inconsistent assumption). Let us consider the problem of information integration from multiple sources which may be mutually contradictory. This situation is common, as the sources are independent, so contradictions may arise. While querying such integration systems it may happen that some sources would be temporarily unreachable (e.g. connection problems, etc) so some inconsistencies may have been omitted from the result to a query. If the use of consistent integrated information (that is, the information on which the sources agree) is essential for the application, then a solution would be to compute and use the part of the answer which is safely consistent. We can do this by considering the worst case, i.e. by assuming that the part of the answer based on unreachable sources is to be considered inconsistent, and rely only on the part that remains consistent. That is, if in doubt we privilege the inconsistency.

Considering our previous example, if we choose the inconsistent assumption then we get $suspect(John)$ is true, $innocent(John)$, $free(John)$, $charge(John)$ and all the other ground atoms are all inconsistent.

The basis of (default) negative information in the well-founded semantics is given by the so-called unfounded sets [19]. We extend this concept by considering as default value for underivable atoms any element of Belnap's four-valued logic [4]: true, false, unknown and inconsistent. Thus we make an optimistic, pessimistic, skeptical and inconsistent assumption, respectively, that will be incorporated elegantly in the definition of the well-founded semantics. Apart the generalization, the difference between the definition in [19] and ours is that the first one has rather a syntactic flavour, while the second has a semantic flavour. Expressing this concept in a semantic manner allows an elegant extension.

As our previous discussion shows, the logic we will consider contains at least four logical values, corresponding to the four mentioned assumptions. However, in fact many real life situations require processing of imperfect information, that is incomplete, inconsistent, and/or uncertain/imprecise. The use of multivalued logics to express the imperfection of information may be needed. In order to illustrate this idea we use the following example.

Suppose we combine information from two sources that are the experts E_1 and E_2 which express their opinion on a statement A. It may be that the two experts are not sure about the truthness of A, due to the imperfection of the available knowledge. The first expert may believe that A is true with a degree of 0.6 of confidence (so there is a degree of 0.4 of doubt), while the second expert may believe that A is true with a degree of 0.8 of confidence (so there is a degree of 0.2 of doubt). If we want to combine the information obtained from the two experts, a natural way would be to consider the consensus of their beliefs: A is true with a degree of confidence of 0.6, and a degree of doubt of 0.2. That is, the pair $\langle 0.6, 0.2 \rangle$ would express the maximal confidence and doubt the two experts agree on. We see such pairs of reals between 0 and 1, expressing degrees of confidence and doubt (note that they are not necessarily complementary w.r.t. 1), as logical values, and we call the space of these logical values the *confidence-doubt logic* - let us denote it by $\mathcal{L}^{\mathcal{CD}}$. We have two orders on $\mathcal{L}^{\mathcal{CD}}$, namely the truth and knowledge orders denoted $\leq_t$ and $\leq_k$, respectively, defined as follows: $\langle x, y \rangle \leq_t \langle z, w \rangle$ iff $x \leq z$ and $w \leq y$, and $\langle x,y\rangle \leq_k \langle z,w\rangle$ iff $x \leq z$ and $y \leq w$, where $\leq$ is the usual order between reals. Intuitively speaking, an increase in the truth order corresponds to an increase in the degree of confidence and a decrease in the degree of doubt, while an increase in the knowledge order corresponds to an increase in both the degree of confidence and the degree of doubt. The least and greatest elements under $\leq_t$ are $\langle 0,1\rangle$ and $\langle 1,0\rangle$, representing no confidence, full doubt, and full confidence, no doubt, respectively. They may be identified with the classical logical values $false$ and $true$. The least and greatest elements under $\leq_k$ are $\langle 0,0\rangle$ and $\langle 1,1\rangle$, representing no confidence, no doubt, and full confidence, full doubt, respectively. They may be identified with the logical values $unknown$ (denoted $\bot$) and $inconsistent$ (denoted $\top$).

Note that $\mathcal{L}^{\mathcal{CD}}$ has an interesting double algebraic structure of lattice (given by the two orders). Such a structure is captured by the concept of bilattice [13]. Bilattices will be used here as multivalued logics in which we define the extended well-founded semantics of extended logic programs. The four assumptions to be considered correspond to a parameter α whose value can be $true$, $false$, $\bot$ or $\top$(which, as we have seen in the example of the confidence-doubt logic, are the extreme values of the bilattice). Once fixed, the value of α represents the "default value" for those atoms of a program that cannot be inferred from the rules. If we want to work under a particular assumption, we choose the appropriate value for α, namely $true$ for the optimistic assumption, $false$ for the pessimistic assumption, $\bot$ for the skeptical assumption and $\top$ for the inconsistent assumption.

We show that, for the pessimistic assumption our extended well-founded semantics captures the conventional well-founded semantics [19] and one of the Fitting's multivalued stable models [8], while for the skeptical assumption our semantics captures the Kripke-Kleene semantics [6].

The paper is organized as follows. In Section 2 we define the extended programs in multivalued logics expressed as bilattices. In Section 3 we define our extended well-founded semantics providing a method to compute it, and we show that it can be obtained in polynomial time with respect to the size of the set of facts from the program. Finally we present related work and concluding remarks in Section 4.

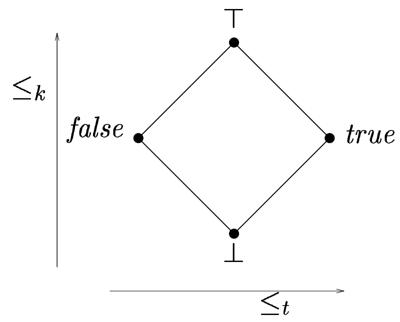

Figure 1: The logic $\mathcal{FOUR}$

2 Preliminaries

2.1 Bilattices

If we consider the four extreme logical values from the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$ presented in the introduction, then we get Belnap's four-valued logic [4], called $\mathcal{FOUR}$, which is depicted in Figure 1. The horizontal axis shows an increase in the degree of truth, while the vertical axis shows an increase in the degree of knowledge. As seen above, the confidence-doubt and Belnap's logics have an interesting algebraic structure of double lattice w.r.t. the truth and knowledge orders. This structure is captured by the concept of bilattice introduced in [13], defined as follows.

Definition 1 *A bilattice is a triple* $\langle \mathcal{B}, \leq_t, \leq_k \rangle$, *where* $\mathcal{B}$ *is a nonempty set, and* $\leq_t$ *and* $\leq_k$ *are partial orders each giving* $\mathcal{B}$ *the structure of a lattice with a least and greatest elements.*

For the bilattice $\mathcal{B}$, join and meet under $\leq_t$ are denoted $\vee$ and $\wedge$ (called extended disjunction and conjunction), and join and meet under $\leq_k$ are denoted $\oplus$ and $\otimes$ (called gullibility and consensus). The greatest and least elements under $\leq_t$ are denoted $true$ and $false$, and the greatest and least elements under $\leq_k$

are denoted $\top$ and $\bot$. Note that the operations $\vee, \wedge, \oplus$ and $\otimes$ are monotone w.r.t. the truth and knowledge orders.

A bilattice has a negation, denoted $\neg$, if $\neg$ is a unary operation which is antimonotone w.r.t. the truth order and monotone w.r.t. the knowledge order. In addition $\neg true = false$, $\neg false = true$, $\neg\bot = \bot$ and $\neg\top = \top$. Note that $\neg$ is an extension of the negation in the two-valued logic.

Note that the binary operations (taking into account the two orders) and the negation of the bilattice $\mathcal{L}^{\mathcal{CD}}$ are given as follows:

$\langle x, y\rangle \ \wedge \ \langle z, w\rangle = \langle min(x, z), max(y, w)\rangle$,
$\langle x, y\rangle \ \vee \ \langle z, w\rangle = \langle max(x, z), min(y, w)\rangle$,
$\langle x, y\rangle \ \otimes \ \langle z, w\rangle = \langle min(x, z), min(y, w)\rangle$,
$\langle x, y\rangle \ \oplus \ \langle z, w\rangle = \langle max(x, z), max(y, w)\rangle$ and
$\neg\langle x, y\rangle = \langle y, x\rangle$.

Note that the operation used in the introduction to combine the two experts' opinions is $\otimes$, that is the consensus.

A bilattice is said to be *distributive* if all the distributive laws built with the extended conjunction and disjunction, consensus and gullibility, hold. Note that the bilattices $\mathcal{FOUR}$ and $\mathcal{L}^{\mathcal{CD}}$ are distributive.

We use bilattices as spaces of logical values for the extended programs we define in the next subsection.

We introduce the concept of *limited* bilattice, used when we evaluate the complexity of the evaluation of the semantics we will introduce.

Definition 2 *The bilattice $\mathcal{B}$ is limited if there exists a polynom p such that for any set of elements $A = \{a_1, \ldots, a_n\}$ from $\mathcal{B}$, the closure of A w.r.t. the bilattice operations has no more than $p(n)$ elements.*

A trivial subclass of limited bilattices is that of the finite bilattices, obviously. However, the limited bilattices class contains also infinite bilattices, as the following proposition shows:

Proposition 1 *The confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$ is a limited bilattice.*

2.2 Extended programs

Conventional logic programming has the set $\{false, true\}$ as its intended space of truth values, but since not every query may produce an answer, partial models are often allowed (i.e. $\bot$ is added). If we want to deal with inconsistency as well, then $\top$ must be added. Fitting extended the notion of logic program, that we will call *extended program*, to bilattices as follows. Let $\mathcal{B}$ be a distributive bilattice with negation.

Definition 3 *[7]*
- A formula is an expression built up from literals and elements of $\mathcal{B}$, using $\wedge, \vee, \otimes, \oplus, \neg, \exists, \forall$.
- A clause or rule r is of the form $P(x_1, ..., x_n) \leftarrow \phi(x_1, ..., x_n)$ where the atomic formula $P(x_1, ..., x_n)$ is the head, denoted by $head(r)$, and the formula $\phi(x_1, ..., x_n)$ is the body, denoted by $body(r)$. It is assumed that the free variables of the body are among $x_1, ..., x_n$.
- A program is a finite set of clauses with no predicate letter appearing in the head of more than one clause (this apparent restriction causes no loss of generality) .

Example 1 *Let P be the following program considered in the context of the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$, where all the atoms are ground:*

$A \leftarrow (C \oplus D) \vee B;$	$B \leftarrow \neg C;$
$D \leftarrow D \otimes \neg A;$	$E \leftarrow \neg C \otimes \neg F;$
$C \leftarrow C \wedge E;$	$F \leftarrow \langle 0.7, 0.1\rangle.$

Note that the rule defining the atom F is a fact. Here F will be assigned the logical value expressing the grades of 0.7 confidence and 0.1 doubt.

A *conventional logic program* [7] is one whose underlying truth-value space is the bilattice $\mathcal{FOUR}$ and which does not involve $\otimes, \oplus, \forall, \bot, \top$.

3 Extended well-founded semantics of extended programs

In the remaining of this paper, in order to simplify the presentation, we assume that all extended programs are instantiated programs, called simply programs.

3.1 Interpretations

We can extend the two orders on bilattice $\mathcal{B}$ to the set of all interpretations over $\mathcal{B}$, denoted by $\mathcal{V}(\mathcal{B})$. An interpretation I of a program P is defined as a partial function over the Herbrand base $\mathcal{HB}_P$, and a completion of I is any total interpretation I' such that $I(A) = I'(A)$, for any atom A in the domain of definition of I, denoted by $def(I)$. When comparing interpretations, we consider their least completion. The least completion of an interpretation I is defined to be the completion J of I such that $J(A) = \bot$, for every atom A not defined under I.

Definition 4 *Let I_1 and I_2 be two interpretations having the least completions I_1' and I_2', respectively. Then $I_1 \leq_t I_2$ if $I_1'(A) \leq_t I_2'(A)$ for all ground atoms A (and similarly for $\leq_k$).*

The total interpretations can be extended from atoms to formulas as follows: $I(X \wedge Y) = I(X) \wedge I(Y)$ (and similarly for the other bilattice operations), $I((\exists x)\phi(x)) = \bigvee_{t \in GT} I(\phi(t))$, and $I((\forall x)\phi(x)) = \bigwedge_{t \in GT} I(\phi(t))$, where GT stands for the set of all ground terms.

However we are interested to see now how *partial* interpretations can be used to evaluate formulas. If B is a closed formula then we say that B evaluates to the logical value β with respect to a partial interpretation I, denoted by $B \equiv \beta$ w.r.t. I, or by $B \equiv_I \beta$, if $J(B) = \beta$ for any completion J of I. The following lemma provides an efficient method of testing whether $B \equiv_I \beta$ by computing the logical value of the formula B w.r.t. only two completions of the interpretation I.

Lemma 1 *Let $I_\bot$ and $I_\top$ be two completions of I defined as follows: $I_\bot(A) = \bot$ and $I_\top(A) = \top$ for every atom A of $\mathcal{HB}_P$ not in $def(I)$. Then $B \equiv_I \beta$ iff $I_\bot(B) = I_\top(B) = \beta$.*

We use also the concept of compatibility of interpretations, defined naturally by:

Definition 5 *The interpretations I and J are said to be compatible if, for any atom A, $I(A)$ and $J(A)$ are both defined implies $I(A) = J(A)$.*

3.2 Semantics of extended programs

Given a program P, we consider two ways of inferring new information from P. First by activating the rules of P and deriving new information through an immediate consequence operator T. Second, by a kind of default reasoning based on the assumption we make in each of the optimistic, pessimistic, skeptical and inconsistent approaches, respectively.

The immediate consequence operator T that we use takes as input an interpretation I and returns an interpretation $T(I)$, defined as follows: for all ground atoms A,

$T(I)(A) = \beta$ if ($A \leftarrow B \in P$ and $B \equiv_I \beta$), and is undefined, otherwise.

Example 2 below illustrates the computation involved in one application of the operator T.

Each assumption is expressed as a hypothesis H^α which formally is an interpretation I that assigns the value α (for $\alpha = true, false, \bot$ and $\top$) to every atom of its domain of definition $def(I)$. Roughly speaking, the hypothesis concerns some of the atoms of the Herbrand base whose logical values cannot be inferred by rule activation.

Definition 6 *Let P be a program and I a partial interpretation. A hypothesis H^α is called sound (w.r.t. P and I) if the following hold:*

1. *H^α is compatible with I and*
2. *for every atom A in $def(H^\alpha)$, if there is a rule r of P with $head(r) = A$ then $body(r) \equiv \alpha$ w.r.t. $I \cup H^\alpha$.*

Intuitively the above definition says that a sound hypothesis must succeed in being tested against the sure knowledge provided by the rules of P (condition (2)) and by a given fixed interpretation I (condition (1)). Note also that if we restrict our attention to conventional logic programs, then the concept of sound hypothesis for $\alpha = false$ reduces to that of unfounded set of Van Gelder et al. [19]. The difference is that the definition in [19] has rather a syntactic flavour, while ours has a semantic flavour. Moreover, our definition not only extends the concept of unfounded set to multivalued logics, but also generalizes its definition w.r.t. the optimistic, pessimistic, skeptical and inconsistent assumptions (corresponding to $\alpha = true, false, \bot$ and $\top$, respectively).

As we explained earlier, given a program P and a partial interpretation I, we derive information in two manners: by activating the rules (i.e. by applying the immediate consequence operator to I) and by making a hypothesis H^α (obtained from one of the four assumptions) and testing it against P and I. If H^α passes the test then it is sound and the information represented by H^α can be derived. In the whole, the information that we derive comes from $T(I)$ and H^α, which are compatible interpretations, as the following proposition states.

Proposition 2 *If H^α is sound w.r.t. the program P and the interpretation I then $T(I)$ and H^α are compatible.*

We note that, for any given P, I and α, there is at least one sound hypothesis H^α (the everywhere undefined interpretation), thus the set of sound hypotheses is nonempty. The following lemma shows that the union of two sound hypotheses is a sound hypothesis:

Lemma 2 *If H_1^α and H_2^α are sound hypotheses w.r.t. an interpretation I, so is their union $H_1^\alpha \cup H_2^\alpha$.*

In fact, it is straightforward to extend this lemma to the union of any set of sound hypotheses w.r.t. I. Therefore the class of sound hypotheses has a greatest element which is obtained by the union of all sound hypotheses H^α w.r.t. I, that we denote by $H^\alpha_{max}(I)$:

Proposition 3 *Let P, I and α be fixed. Then there is a sound hypothesis $H^\alpha_{max}(I)$ such that: $T(I) \cup H^\alpha \leq_k T(I) \cup H^\alpha_{max}(I)$, for all sound hypotheses H^α w.r.t. I.*

Let us illustrate the concept through the example below.

Example 2 *Let P be the program considered in Example 1 in the context of the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$ and let us consider the interpretation:*

$$I = \left[\begin{array}{cccccc} A & B & C & D & E & F \\ \langle 1,0\rangle & \langle 1,0\rangle & \langle 0,1\rangle & & & \langle 0.7,0.1\rangle \end{array} \right]$$

where the Herbrand base atoms are displayed on the first row, and their corresponding logical values on the second row. Note that D and E are undefined under I, as no value corresponds to them. Here A and B are interpreted as true (i.e. 1.0 confidence and 0 doubt), C is interpreted as false (i.e. 0 confidence and 1.0 doubt) while F is associated with the grades of 0.7 confidence and 0.1 doubt.

If we wish to evaluate $T(I)(A)$ we have to evaluate the formula $U = (C \oplus D) \vee B$ w.r.t. I. Note that D is undefined under I, so we apply the least and greatest completions of I to U (according to Lemma 1), and get: $I_\bot(U) = \langle 1,0\rangle$ and $I_\top(U) = \langle 1,0\rangle$, so $U \equiv \langle 1,0\rangle$ w.r.t. I and thus $T(I)(A) = \langle 1,0\rangle$.

Similarly, $T(I)(D)$ is computed by trying to evaluate the formula $V = D \otimes \neg A$ w.r.t. I. We have $I_\bot(V) = \langle 0,0\rangle \otimes \neg\langle 1,0\rangle = \langle 0,0\rangle \otimes \langle 0,1\rangle = \langle 0,0\rangle$; then $I_\top(V) = \langle 1,1\rangle \otimes \neg\langle 1,0\rangle = \langle 1,1\rangle \otimes \langle 0,1\rangle = \langle 0,1\rangle$. As the two values are different, $T(I)(D)$ remains undefined. Continuing similarly, we get the interpretation

$$T(I) = \left[\begin{array}{cccccc} A & B & C & D & E & F \\ \langle 1,0\rangle & \langle 1,0\rangle & & & \langle 0.1,0\rangle & \langle 0.7,0.1\rangle \end{array} \right]$$

Now, let us consider the logical value $\alpha = \langle 0,1\rangle$ (i.e. false), and let us try to construct the greatest sound hypothesis $J = H^\alpha(I)$. According to Definition 6 A, B and F have to be undefined under J, otherwise J would not be compatible with I.

If we consider the remaining atoms we would have

$$J = \left[\begin{array}{cccccc} A & B & C & D & E & F \\ & & \langle 0,1\rangle & \langle 0,1\rangle & \langle 0,1\rangle & \end{array} \right]$$

However, J is not sound since, according to point 2 from Definition 6, the body of the rule with the head E should evaluate to $\langle 0,1\rangle$ w.r.t. $I \cup J$, and we see it evaluates to $\langle 0.1,0\rangle$.

Now if we consider the remaining atoms we have

$$J = \left[\begin{array}{cccccc} A & B & C & D & E & F \\ & & \langle 0,1\rangle & \langle 0,1\rangle & & \end{array} \right]$$

If, again, we evaluate the bodies of the rules with the heads C and D w.r.t. $I \cup J$ we get $\langle 0,1\rangle$, so J is sound. By construction J is maximal with this property, so it is the greatest sound hypothesis w.r.t. I.

We have seen that there are two ways of deriving information from a program: By applying the rules and by using sound hypotheses (namely the maximal ones, for deriving all possible information). Hence the idea to build the following sequence, where I^u is the everywhere undefined interpretation:

$I_0 = I^u$
$I_{i+1} = T(I_i) \cup H^{\alpha}_{max}(I_i)$
$I_j = \bigcup_{i<j} I_i$ if j is a limit ordinal.

We have:

Proposition 4 *The sequence $I_{i\geq 0}$, is increasing w.r.t. the knowledge order, and has a limit denoted $lfp^{\alpha}(P)$.*

Note that, intuitively speaking, $lfp^{\alpha}(P)$ represents all the information that can be inferred from the program P. Obviously it is a fixpoint of the operator $T \cup H^{\alpha}_{max}$. Moreover, we ca show that $lfp^{\alpha}(P)$ has an important property namely that it satisfies the rules of the program P.

Definition 7 *An interpretation I is a model of a program P if for every rule $A \leftarrow B$ of P, $I(B) \leq_t I(A)$.*

This definition comes from the intuitive remark that, as the consequence is derived from a premise, the degree of truth of the consequence should be at least the degree of truth of the premise.

Proposition 5 *The interpretation $lfp^{\alpha}(P)$ is a model of P.*

This justifies the following definition of semantics for P.

Definition 8 *The interpretation $lfp^{\alpha}(P)$ is defined to be the extended well-founded semantics of P w.r.t. the logical value α, that we denote by $ewfs^{\alpha}(P)$.*

Considering the four different assumptions, we have the following relationships between the semantics obtained:

Proposition 6 *If P is a program then:*

$ewfs^{\perp}(P) \leq_k ewfs^{\alpha}(P) \leq_k ewfs^{\top}(P)$

for $\alpha \in \mathcal{FOUR}$.

Fitting [8] introduced the (multivalued) stable models for extended programs in bilattices. This concept extends that of stable model in the conventional bivalued logic [12]. We show that if we consider the pessimistic approach, our semantics coincides with Fitting's (multivalued) stable model that has the least degree of information.

Theorem 1 *Let P be an extended program and $mstable_k(P)$ be its least multivalued stable model w.r.t. $\leq_k$, as defined in [8]. Then $ewfs^{false}(P) = mstable_k(P)$.*

This equality may seem surprising since $ewfs^{false}(P)$ advantages negative information while $mstable_k(P)$ prefers the lack of information, as it is minimal in the knowledge order. However, as Fitting shows in [8, 9], in the definition of (multivalued) stable models there exists a preference for falsehood, in the sense that, whenever possible, the truth value *false* is assigned to atoms. See [9] for an approach of building a theory that prefers falsehood.

The last result of the subsection compares our semantics with the well-founded semantics [19] and Kripke-Kleene semantics [6] of a conventional program P, denoted $wfs(P)$ and $kks(P)$ respectively.

Theorem 2 *If P is a conventional program and the bilattice is $\mathcal{FOUR}$ then $ewfs^{false}(P) = wfs(P)$ and $ewfs^{\perp}(P) = kks(P)$.*

3.3 Computing the extended well-founded semantics

We now give a method for computing the greatest sound hypothesis H^{α}_{max} used in the definition of the semantics we have introduced.

Given the interpretation I, consider the following sequence $\langle PF_i(I)\rangle$, $i \geq 0$:

$PF_0(I) = \emptyset$;

$PF_{i+1}(I) = \{A \mid A \leftarrow B \in P$ and $B \not\equiv \alpha$ *w.r.t.* $J_{i,I}\}$, for $i \geq 0$, where $J_{i,I}$ is the interpretation defined by:

$$J_{i,I}(A) = \begin{cases} I(A) \text{ if } A \in def(I), \\ \alpha \text{ if } A \in (\mathcal{HB}_P \setminus PF_i) \setminus def(I), \\ undefined, otherwise. \end{cases}$$

We have the following results:

Proposition 7 *The sequence $\langle PF_i(I)\rangle$, $i \geq 0$ is increasing with respect to set inclusion and it has a limit, denoted $PF(I)$.*

Theorem 3 *Let P, I and α be fixed. If J is an interpretation defined by : $J(A) = \alpha$ for any $A \in (\mathcal{HB}_P \setminus PF(I))$ and $J(A) = undefined$ for any other ground atom A, then $H^{\alpha}_{max}(I) = J$.*

We note that, if we restrict our attention to conventional programs in the logic $\mathcal{FOUR}$ and the pessimistic approach (i.e. $\alpha = false$), the set $PF(I)$ corresponds to the set of potentially founded facts of [3].

We illustrate the computation of the extended well-founded semantics through the following example:

Example 3 *Let P be the program considered in Example 1 in the context of the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$.*

Let $\alpha = \langle 0, 1\rangle$ (i.e. a pessimistic assumption). The following sequence of interpretations is obtained by iterating the operator $T \cup H^{\alpha}_{max}$, starting with the everywhere undefined interpretation $I_0 = I^u$. Note that the greatest sound hypothesis, as a part of each I_i, is indicated in bold, and it is computed using the method presented above.

$$I_1 = \left[\begin{array}{cccccc} A & B & \mathbf{C} & D & E & F \\ & & \langle \mathbf{0}, \mathbf{1}\rangle & & & \langle 0.7, 0.1\rangle \end{array} \right]$$

$$I_2 = \left[\begin{array}{cccccc} A & B & \mathbf{C} & D & E & F \\ & \langle 1, 0\rangle & \langle \mathbf{0}, \mathbf{1}\rangle & & \langle 0.1, 0\rangle & \langle 0.7, 0.1\rangle \end{array} \right]$$

$$I_3 = \left[\begin{array}{cccccc} A & B & \mathbf{C} & D & E & F \\ \langle 1, 0\rangle & \langle 1, 0\rangle & \langle \mathbf{0}, \mathbf{1}\rangle & & \langle 0.1, 0\rangle & \langle 0.7, 0.1\rangle \end{array} \right]$$

$$I_4 = \left[\begin{array}{cccccc} A & B & \mathbf{C} & \mathbf{D} & E & F \\ \langle 1, 0\rangle & \langle 1, 0\rangle & \langle \mathbf{0}, \mathbf{1}\rangle & \langle \mathbf{0}, \mathbf{1}\rangle & \langle 0.1, 0\rangle & \langle 0.7, .1\rangle \end{array} \right]$$

If we compute I_5 we get $I_5 = I_4$, so I_4 represents both the whole information that can be derived from P making assumptions w.r.t. $\alpha = \langle 0, 1\rangle$ and the semantics of P. So in the semantics of the program A and B are true, C and D are false, E is associated with the grades of 0.1 confidence and 0 doubt while F is associated with the grades of 0.7 confidence and 0.1 doubt.

If we compute the semantics of the program P provided in the introduction, for each of the pessimistic, optimistic, skeptical and inconsistent approaches respectively, we get the following table, where we have included on the first row only the ground atoms built with predicates defined by the program rules[1]:

Table1: $ewfs^{\alpha}(P)$

α	s(John)	i(John)	f(John)	c(John)
false	*true*	*false*	*false*	*true*
true	*true*	*true*	*true*	*false*
$\bot$	*true*	$\bot$	$\bot$	$\bot$
$\top$	*true*	$\top$	$\top$	$\top$

We conclude this section with a complexity result showing that our semantics can be computed in polynomial time with respect to the size of the set of facts from the program.

Formally, let $\mathcal{B}$ be a limited bilattice and let $P = P_{Rules} \cup P_{Facts}$ be a program with no function symbol, where P_{Facts} is the set of facts (i.e. the set of rules of the form $A \leftarrow c$ where c is a logical value from the bilattice $\mathcal{B}$) and P_{Rules} is the set of rules (i.e. the remaining part of P). Note that as the program is function free, the fixpoint computation of our semantics terminates in a finite number of steps.

Theorem 4 *The time complexity of the computation of the extended well-founded semantics of the program P in a limited bilattice is polynomial w.r.t. $|P_{Facts}|$.*

4 Related work and concluding remarks

We have proposed an approach for handling imperfect information in logic programs by defining the extended well-founded semantics. We consider imperfect information to be missing/incomplete, uncertain and/or inconsistent. In our semantics, the missing information is completed by using *optimistic, pessimistic, skeptical* and *inconsistent* assumptions. The imperfect information is handled by using bilattices as multivalued logics. We provide a method of computation of our semantics and show that, for the pessimistic as-

[1]All the other ground atoms are assigned the corresponding logical value α, and have been omitted.

sumption our extended well-founded semantics captures the conventional well-founded semantics and Fitting's least multivalued stable model, while for the skeptical assumption our semantics captures the Kripke-Kleene semantics.

The conventional logic program semantics are mostly based on pessimistic and skeptical approaches [6, 12, 18, 19], which is also the case with the most extended semantics expressing uncertainty [21, 17, 15, 14, 8, 10]. Although useful - as we have shown through our example in introduction, the optimistic approach has been uncommon. In [9] Fitting proposes a mechanism of building a theory of truth that prefers falsehood, that is, the logical value $false$ is assigned to atoms whenever possible. It is suggested that this mechanism, used also for defining the (multivalued) stable models for extended programs [8], can be straightforwardly extended to build a theory that prefers truthhood, which would correspond, in our terms, to using an optimistic assumption. We have shown the connection between our semantics defined in the pessimistic approach and the least (multivalued) stable model.

Our semantics can express also total or partial inconsistency. For instance if we consider the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$ total inconsistency refers to the logical value $\langle 1, 1\rangle$, while partial inconsistency to logical values as $\langle 0.6, 0.8\rangle$ where the sum of the two degrees of confidence and doubt is more than 1. These inconsistencies may emerge from combining contradictory information in the program as for instance $\langle 1, 0\rangle \oplus \langle 0, 1\rangle = \langle 1, 1\rangle$, or by using an inconsistent assumption. Note that one characteristic of our semantics is that inconsistency is pinpointed, i.e. it does not lead to entailment of any conclusion. That is, our semantics is paraconsistent. [1] proposes a declarative semantics for programs with conventional negation and negation by default, which deals with inconsistency. This semantics is paraconsistent as our semantics is, however it is limited in expressing only information which is true, false, undefined or inconsistent, and not having other grades of uncertainty. [5] provides an excellent survey of paraconsistent semantics of logic programs. In [11] one considers ordered logic programs with negative heads, and two approaches of resolving conflict between rules leading to different notions of program models limited to the three-valued logic. [11] defines three-valued (or partial) models, in particular the credulous and sceptic models as a maximal and minimal c-assumption free c-partial models, w.r.t. set inclusion (that is, they represent a maximal and a minimal degree of information). [11] shows how these concepts relate to those of well-founded model and stable models, but no comparison is made with the skeptical Kripke-Kleene semantics. One of differences w.r.t. our approach is that in [11] inconsistency is avoided by ignoring (conflicting) rules, while we allow inconsistency, total or partial, as part of imperfect information represented in programs, and rules are preserved. On the other hand imperfect information in [11] resumes only to missing information.

We have also shown that the complexity of the evaluation of the extended well-founded semantics with respect to the size of the set of program facts is polynomial time, if we consider a useful class of bilattices which may be finite or infinite, namely the limited bilattices, containing also the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$ and the four valued logic $\mathcal{FOUR}$. We are currently investigating the possibility of using our approach in the area of intelligent agents used for retrieval and integration of imperfect information.

References

[1] ARIELI O. *Paraconsistent declarative semantics for extended logic programs*, Annals of Mathematics and Artificial Intelligence 36(4), 381-417, 2002

[2] ARIELI, O. and AVRON, A. *The Logical Role of the Four-Valued Bilattice*, Proc. 13th Annual IEEE Symp. on Logic in Computer Science, 118-126, 1998

[3] BIDOIT N., FROIDEVEAUX C., *Negation by default and unstratifiable logic programs*, TCS, 78, 1991

[4] BELNAP, N. D., Jr, *A Useful Four-Valued Logic*, in: J. M. Dunn and G. Epstein (eds.), Modern Uses of Multiple-valued Logic, D. Reichel, Dordrecht, 1977.

[5] DAMASIO, C. and PEREIRA, L., *A survey of paraconsistent semantics for logic programas*, in: D. Gabbay and P. Smets (eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 2, Kluwer, 1998.

[6] FITTING, M. C., *A Kripke/Kleene Semantics for Logic Programs*, J. Logic Programming, 2:295-312, 1985

[7] FITTING, M. C., *Bilattices and the Semantics of Logic Programming*, J. Logic Programming, 11:91-116, 1991

[8] FITTING, M. C., *The Family of Stable Models*, J. Logic Programming, 17:197-225, 1993

[9] FITTING, M. C., *A theory of truth that prefers falsehood*, Journal of Philosophical Logic , 26:477–500, 1997

[10] FITTING, M. C., *Fixpoint semantics for logic programming - a survey*, Theoretical Computer Science, 278:25-51, 2002

[11] GABBAY D. M., LAENENS E., VERMEIR D., *Credulous vs. Sceptical Semantics for Ordered Logic Programs*, KR 1991, 208-217, 1991

[12] GELFOND, M., and LIFSCHITZ V., *The Stable Model Semantics for Logic Programming*, Proc. of the 5th International Conference on Logic Programming, 1988

[13] GINSBERG, M. L., *Multivalued Logics: a Uniform Approach to Reasonning in Artificial Intelligence*, Computationnal Intelligence, 4:265-316, 1988.

[14] LAKSHMANAN V. S. and SADRI F., *Modeling Uncertainty in Deductive Databases*, Proc. of 5th International Conference of Database and Expert Systems Applications, 1994

[15] LAKSHMANAN V. S. and SADRI F., it On a theory of probabilistic deductive databases, J. Theory and Practice of Logic Programming, 1(1), 2001

[16] LOYER Y., SPYRATOS N. and STAMATE D. *Hypothesis-based semantics of logic programs in multivalued logics*, ACM Transactions on Computational Logic 15(3), 508-527, 2004

[17] NG R. and SUBRAHMANIAN V. S. *Probabilistic Logic Programming*, Journal of Information and Computation, 101:2, 1992

[18] PRZYMUSINSKI, T. C. *The well-founded semantics coincides with the three-valued stable semantics*, Fundamenta Informaticae, 13(4):445-464, 1990

[19] VAN GELDER, A., ROSS, K. A., SCHLIPF, J. S., *The Well-Founded Semantics for General Logic Programs*, J. ACM, 38:620-650, 1991

[20] SIM K. M. *Bilattices and Reasoning in Artificial Intelligence: Concepts and Foundations* Artificial Intelligence Review 15:3, 219-240, 2001

[21] SUBRAHMANIAN V. S. , *Probabilistic Databases and Logic Programming*, Proc. of 17th International Conference of Logic Programming, 2001