International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:9, 2007

# Extended Well-Founded Semantics in Bilattices

Daniel Stamate

***Abstract*****—** 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 from the programs are considered to be false (i.e. a pessimistic assumption). One of the most successful semantics of conventional logic programs based on the CWA is the well-founded semantics. However, the CWA is not applicable in all circumstances when information is handled. That is, the well-founded semantics, if conventionally defined, would behave inadequately in different cases. The solution we adopt in this paper is to extend the well-founded semantics in order for it to be based also on other assumptions. The basis of (default) negative information in the well-founded semantics is given by the so-called unfounded sets. We extend this concept by considering optimistic, pessimistic, skeptical and paraconsistent assumptions, used to complete missing information from a program. Our semantics, called extended well-founded semantics, expresses also imperfect information considered to be missing/incomplete, uncertain and/or inconsistent, by using bilattices as multivalued logics. We provide a method of computing the extended well-founded semantics and show that Kripke-Kleene semantics is captured by considering a skeptical assumption. We show also that the complexity of the computation of our semantics is polynomial time.

***Keywords*****—**Logic programs, imperfect information, multivalued logics, bilattices, assumptions.

## I. 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 [15]. 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 through the following legal case example:

D. Stamate is with the Department of Computing, Goldsmiths College, University of London, SE146NW London, UK (phone +44-2079197864; fax +44-2079197853; e-mail: d.stamate@doc.gold.ac.uk).

$P$ : charge(X) $\leftarrow$ suspect(X) $\wedge \neg$innocent(X)
free($X$) $\leftarrow$ suspect(X) $\wedge$ innocent(X)
innocent(X) $\leftarrow$ $\exists$Y(alibi(X,Y) $\wedge \neg$relatives(X,Y))
suspect(John) $\leftarrow$ $true$

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. a paraconsistent 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 [15]. We extend this concept by considering as default value for underivable atoms any element of Belnap's four-valued logic [3]: true, false, unknown and inconsistent. Thus we make an optimistic, pessimistic, skeptical and paraconsistent assumption, respectively, that will be incorporated elegantly in the definition of the well-founded semantics. Apart the generalization, the difference between the definition in [15] 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. 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 [9]. 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 $\alpha$ 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 $\alpha$ 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 $\alpha$, namely $true$ for the optimistic assumption, $false$ for the pessimistic assumption, $\bot$ for the skeptical assumption and $\top$ for the paraconsistent assumption.

We show that, for the pessimistic assumption our extended well-founded semantics captures the conventional well-founded semantics [15], while for the skeptical assumption our semantics captures the Kripke-Kleene semantics [5].

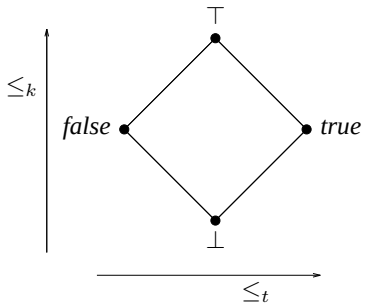


Fig. 1 The logic $\mathcal{FOUR}$

Our formalism of defining the semantics is paraconsistent, in the sense that in our semantics inconsistent information does not entail every conclusion, but it is localised.

The main motivation comes from the area of knowledge acquisition, where on the one hand the knowledge may be imperfect (incomplete, uncertain, etc), and on the other hand contradictions may occur during the process of collecting knowledge from different experts. Indeed, in multi-agent systems, different agents may give different answers to the same query. It is then important to be able to process the answers so as to extract the maximum of information on which the various agents agree, or to detect the items on which the agents give conflicting answers. Incompleteness of the knowledge may be resolved by agents by using hypotheses. Thus one may distinguish pessimistic, optimistic and skeptical agents, according to the type of assumption they may use.

The paper is organized as follows. In Section II we define the extended programs in multivalued logics expressed as bilattices. In Section III 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 some concluding remarks and suggestions for further research in Section IV.

## II. Preliminaries

### A. Bilattices

The bilattice approach introduced by M. Ginsberg [9] is an important contribution to the concept of multivalued logics. Bilattices and/or their derived sublogics are useful in expressing uncertainty, incompleteness and inconsistency in logic programming, artificial intelligence and databases [1], [7], [8], [12], [13], [16].

If we consider the four extreme logical values from the confidence-doubt logic $\mathcal{L}^{\mathcal{CD}}$, then we get Belnap's four-valued logic [3], 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, 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 element 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.

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. □

### B. 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:* [6]
- 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) . □

Fitting also defined the family of conventional logic programs. A *conventional logic program* is one whose underlying truth-value space is the bilattice $\mathcal{FOUR}$ and which does not involve $\otimes, \oplus, \forall, \bot, \top$. Such programs can be written in the customary way, using commas to denote conjunction.

## III. 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. Moreover, we use the term "program" to mean "extended program", unless explicitly stated otherwise.

### A. 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 $\beta$ with respect to an 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 reason we consider completions is that if $I$ is an interpretation then some of the atoms of $B$ may not be associated with a logical value under $I$, and therefore we will not be able to evaluate $B$. However, there are formulas $B$ in which these atoms do not matter for the logical value that can be associated to $B$. For example consider $B = A \vee C$ and the interpretation $I$ defined by $I(C) = true$, then no matter how $A$ is interpreted $B$ can be evaluated to $true$. Thus we can write $B \equiv_I true$.

The following lemma provides a 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 $I_\perp$ and $I_\top$ be two completions of $I$ defined as follows: $I_\perp(A) = \perp$ and $I_\top(A) = \top$ for every atom $A$ of $\mathcal{HB}_P$ not in $def(I)$. Then $B \equiv_I \beta$ iff $I_\perp(B) = I_\top(B) = \beta$. □

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 then $I(A) = J(A)$. □

### B. 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 paraconsistent 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 $A$,

$$T(I)(A) = \begin{cases} \beta \text{ if } A \leftarrow B \in P \text{ and } B \equiv_I \beta \\ \text{undefined, otherwise} \end{cases}$$

Each assumption is expressed as a hypothesis $H^\alpha$ which formally is an interpretation $I$ that assigns the value $\alpha$ (for $\alpha = true, false, \perp$ 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. The hypothesis must be tested against the "sure" knowledge provided by the rules of $P$ and by a given fixed interpretation $I$ (which can be the everywhere undefined interpretation). The test consists of "adding" $H^\alpha$ to $P$ and $I$, activating the rules of $P$ and deriving all information that can be derived by $T$. If at the end of this process every atom in $def(H^\alpha)$ is assigned the value $\alpha$, this means that the hypothesis $H^\alpha$ is a sound one, i.e. that the assignment of the value $\alpha$ by $H^\alpha$ is not in contradiction with $P$ and $I$. Hence the following definition:

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

1) $H^\alpha$ 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. $I \cup H^\alpha$.

□

Several remarks are in order here concerning the above definition. First, let us observe that if we restrict our attention to conventional logic programs (recall that the class of extended programs strictly contains the conventional programs), then the concept of sound hypothesis for $\alpha = false$ reduces to that of unfounded set of Van Gelder et al. [15]. The difference is that the definition in [15] 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 paraconsistent assumptions (corresponding to $\alpha = true, false, \perp$ and $\top$, respectively).

Second, the compatibility of $H^\alpha$ with $I$ (point 1 of the definition) implies that $I \cup H^\alpha$ is an interpretation. This is necessary in order to be able to evaluate the bodies of rules in $P$ (point 2 of the definition). We recall that Lemma 1 provides an efficient method for this evaluation.

Finally, let us stress that the bodies of rules are evaluated with respect to completions of $I \cup H^\alpha$ (and not just w.r.t. $I \cup H^\alpha$). This ensures that every atom appearing in a rule body is assigned a logical value before evaluation takes place.

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 an assumption/hypothesis $H^\alpha$ and testing it against $P$ and $I$ (roughly speaking, this test is point 2 of Definition 6). If $H^\alpha$ passes the test then it is sound and the information represented by $H^\alpha$ can be derived. In the whole, the information that we derive comes from $T(I) \cup H^\alpha$, assuming of course that $T(I) \cup H^\alpha$ is indeed an interpretation, i.e. assuming that $T(I)$ and $H^\alpha$ are compatible. The following proposition asserts that this is the case.

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

We note that, for any given $P$, $I$ and $\alpha$, there is at least one sound hypothesis $H^\alpha$ (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 $H_1^\alpha$ and $H_2^\alpha$ 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 $H^\alpha$ w.r.t. $I$, that we denote by $H^\alpha_{max}(I)$:

*Proposition 3:* Let $P$, $I$ and $\alpha$ 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 $H^\alpha$ w.r.t. $I$. □

Now, roughly speaking, the semantics that we would like to associate with a program $P$ is the maximum of information that we can derive from $P$ under a sound hypothesis $H^\alpha$ but without any other information. To implement this idea we proceed as follows:

1) In order to ensure the maximum of derived information we use the greatest sound hypothesis $H^\alpha_{max}(I)$.
2) As we do not want any extra information (other than $P$ and $H^\alpha_{max}(I)$), we use the everywhere undefined interpretation, call it $I^\perp$.

3) In order to actually derive the maximum of information from $P$ and $I^{\perp}$, we apply the operator $T \cup H^{\alpha}_{max}$ iteratively, until fixpoint, starting with $I = I^{\perp}$.

The above idea of computation is formally justified by the following proposition.

*Proposition 4:* The following sequence of interpretations $I_i, i \geq 0$, is increasing w.r.t. knowledge order, and has a limit denoted $lfp^{\alpha}(P)$:
$I_0 = I^{\perp}$
$I_{i+1} = T(I_i) \cup H^{\alpha}_{max}(I_i)$
$I_j = \bigcup_{i<j} I_i$ if j is a limit ordinal. □

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 $\alpha$, 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:
1) $ewfs^{\perp}(P) \leq_k ewfs^{true}(P) \leq_k ewfs^{\top}(P)$ and
2) $ewfs^{\perp}(P) \leq_k ewfs^{false}(P) \leq_k ewfs^{\top}(P)$. □

The last result of the subsection compares our semantics with the conventional well-founded semantics [15] and Kripke-Kleene semantics [5] of a program $P$, denoted $wfs(P)$ and $kks(P)$ respectively.

*Theorem 1:* If $P$ is a conventional program and the bilattice is $\mathcal{FOUR}$ then $ewfs^{false}(P) = wfs(P)$ and $ewfs^{\perp}(P) = kks(P)$. □

### C. Computing the Extended Well-founded Semantics

We now give a method for computing the greatest sound hypothesis $H^{\alpha}_{max}$ 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:

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

Roughly speaking, in the above computation we want to evaluate step by step the atoms that could potentially have a logical value different than $\alpha$ in $ewfs^{\alpha}(P)$. The rest of the atoms in the Herbrand base makes up the domain of definition of $H^{\alpha}_{max}$. 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 2:* Let $P$, $I$ and $\alpha$ 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 [2].

If we compute the semantics of the program $P$ provided in the introduction, for each of the pessimistic, optimistic, skeptical and paraconsistent 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: The extended well-founded semantics of P*

| Hypothesis $H^{\alpha}$ | s(John) | i(John) | f(John) | c(John) |
|---|---|---|---|---|
| any atom is $false$ | $true$ | $false$ | $false$ | $true$ |
| any atom is $true$ | $true$ | $true$ | $true$ | $false$ |
| any atom is $\perp$ | $true$ | $\perp$ | $\perp$ | $\perp$ |
| any atom is $\top$ | $true$ | $\top$ | $\top$ | $\top$ |

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.

[1] All the other ground atoms are assigned the corresponding logical value $\alpha$, and have been ommitted.

*Theorem 3:* The complexity of the computation of the extended well-founded semantics of the program $P$ in a limited bilattice is polynomial w.r.t. $|P_{Facts}|$. □

## IV. 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 *paraconsistent* 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 assumption our extended well-founded semantics captures the conventional well-founded semantics, 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, while the optimistic approach has been uncommon. [4] provides an excellent survey of paraconsistent semantics of logic programs.

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 restrict the multivalued logic we use to a limited bilattice. 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. and Avron, A. *The Logical Role of the Four-Valued Bilattice*, Proc. 13th Annual IEEE Symp. on Logic in Computer Science, 118-126, 1998
[2] Bidoit N., Froideveaux C., *Negation by default and unstratifiable logic programs*, TCS, 78, 1991
[3] 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.
[4] 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.
[5] Fitting, M. C., *A Kripke/Kleene Semantics for Logic Programs*, J. Logic Programming, 2:295-312, 1985
[6] Fitting, M. C., *Bilattices and the Semantics of Logic Programming*, J. Logic Programming, 11:91-116, 1991
[7] Fitting, M. C., *The Family of Stable Models*, J. Logic Programming, 17:197-225, 1993
[8] Fitting, M. C., *Fixpoint semantics for logic programming - a survey*, Theoretical Computer Science, 278:25-51, 2002
[9] Ginsberg, M. L., *Multivalued Logics: a Uniform Approach to Reasoning in Artificial Intelligence*, Computationnal Intelligence, 4:265-316, 1988.
[10] Ginsberg, M. L., *Bilattices and modal operators*, J. of Logic Computation, 1:41-69, 1990.
[11] Loyer, Y., Spyratos, N., and Stamate,D., *Computing and Comparing Semantics of Programs in Four-valued Logics*, in: Proceedings of the 24th Symposium on Mathematical Foundations of Computer Science ($MFCS'99$), LNCS 1672, Springer Verlag, 1999
[12] Loyer Y., Spyratos N. and Stamate D. *Parameterised Semantics for Logic Programs - a Unifying Framework*, Theoretical Computer Science, 308(1-3), 429-447, 2003
[13] 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
[14] Przymusinski, T. C. *The well-founded semantics coincides with the three-valued stable semantics*, Fundamenta Informaticae, 13(4):445-464, 1990
[15] Van Gelder, A., Ross, K. A., Schlipf, J. S., *The Well-Founded Semantics for General Logic Programs*, J. ACM, 38:620-650, 1991
[16] Sim K. M. *Bilattices and Reasoning in Artificial Intelligence: Concepts and Foundations* Artificial Intelligence Review 15:3, 219-240, 2001