Minimal Well-Founded Semantics and Autoepistemic Circumscription [*]

Li Yan Yuan
Department of Computing Science
The University of Alberta
Edmonton, CANADA T6G 2H1
yuan@cs.ualberta.ca

Abstract

In this paper we propose the minimal well-founded semantics for logic programs with negation based on the fixpoint of the double Gelfond-Lifschitz transformation which overcomes the existing problems associated with the stable, the well-founded, and the stable class semantics. By representing logic programs as autoepistemic theories, we are able to represent various semantics as simple circumscription formulas

KEY WORDS: Logic Program Semantics, Autoepistemic Logic, Circumscription

1 Introduction

The fixpoint and the alternating fixpoint, based on the Gelfond-Lifschitz (GL) transformation introduced in [5], have been used to characterize both the stable and the well-founded semantics, two most prominent semantics for logic programs with negation [2, 1]. In fact, an interpretation I is a stable model of program $\mathcal{P}$ if and only if it is a fixpoint of the GL-transformation of $\mathcal{P}$ and the well-founded semantics is determined by the set of all alternating fixpoints, i.e., fixpoints of the double GL transformation, of program $\mathcal{P}$.

[*]This paper is available as a technical report TR92-15 at the University of Alberta, Computing Science Department.

The stable semantics [5] for logic programs is handicapped by the no-stable-model problem that programs may not always have stable models while the well-founded semantics [3] may not be adequate to characterize the intuitive meaning of logic programs [6, 1]. (See Section 3 for details.)

Recently, Baral and Subrahmanian have proposed the stable class semantics to resolve the problems associated with the stable and the well-founded semantics[1]. The basic idea is that the GL transformation of a program may not have any fixpoint, but there might be a collection of points, called the *stable class*, so the GL-transformation cycles around this collection of points. Then the stable class semantics is defined by the union of all minimal stable classes. As shown in [1], the stable class semantics is always consistent and it indeed presents the intuitive meaning for some programs. However, Example 3.3 in Section 3 shows that the stable class semantics suffers from the problem that unreasonable conclusions may be deduced.

In this paper, we are trying to resolve the same problems addressed by Baral and Subrahmanian, but take a different approach. A program may not always have fixpoints, which is the cause of the no-stable-model problem, but it always has at least one alternating fixpoint. Instead of considering stable classes, we concentrate on alternating fixpoints.

Since the well-founded semantics is determined by the set of all alternating fixpoints, not necessarily those minimal ones, and the inadequateness of the well-founded semantics is caused by those non-minimal alternating fixpoints, it is natural to extend the well-founded semantics by using only minimal alternating fixpoints, which leads to our definition of the *minimal well-founded semantics*.

We have demonstrated that the minimal well-founded semantics has overcome those existing problems associated with the stable, the well-founded, and the stable class semantics.

Following the direction of representing a logic program as an autoepistemic theory [4, 10, 11], we are able to represent various semantics of logic programs in terms of second order formulas, namely, circumscription formulas on autoepistemic theories.

Let $\mathcal{P}(P, \mathcal{L}P)$ be an autoepistemic theory for a logic program, we show that the stable semantics is characterized by

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge (P \equiv \mathcal{L}P)$$

and the well-founded semantics is characterized by

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge CIRC(\mathcal{P}(\mathcal{L}P, P); \mathcal{L}P)$$

The minimal well-founded semantics can also be characterized by a second order formula.

2 The Alternating Fixpoint Revisited

A logic program is a set of clauses of the form

$$a \leftarrow b_1, \ldots, b_n, \neg c_1, \ldots, \neg c_m$$

where $m, n \geq 0$ and a, b_i's, and c_j's are atoms. Without loss of generality we assume that the program has been *instantiated* and thus consists of a (possibly infinite) set of *propositional clauses.*

Let $\mathcal{P}$ be a program and I a (two-valued) Herbrand interpretation of $\mathcal{P}$. Then the Gelfond-Lifschitz transformation of $\mathcal{P}$ with respect to I is the logic program P^I obtained from $\mathcal{P}$ as follows:

1. eliminating from $\mathcal{P}$ each clause whose body contains the negation of an atom in I;

2. from the body of each remaining clause in $\mathcal{P}$, delete all negative literals.

Recall the transformation $\mathbf{T}_P(I)$, called the *immediate consequence operator for a Horn program*, whose output is a set of atoms such that $a \in \mathbf{T}_P(I)$ if and only if a is the head of some clause in $\mathcal{P}$ all of whose literals in the body are in I. P^I is a Horn program and hence has a unique least model which is given by $\mathbf{T}_{P^I} \uparrow \omega$.

We define $\mathbf{S}_P(I) = \mathbf{T}_{P^I} \uparrow \omega$. A *fixpoint* of $\mathbf{S}_P$ is an interpretation of $\mathcal{P}$ such that $I = \mathbf{S}_P(I)$. A fixpoint of $\mathbf{S}_P$ is also called a fixpoint of $\mathcal{P}$

Proposition 2.1 ([5, 2]) I is a stable model of $\mathcal{P}$ if and only if I is a fixpoint of $\mathbf{S}_P$. $\Box$

One of the problem here is that $\mathbf{S}_P$ may not always have fixpoints.

Van Gelder proposed the alternating fixpoint to resolve the problem. The basic idea is that $\mathbf{S}_P$ man not always have fixpoints, but $\mathbf{S}_P(\mathbf{S}_P(I))$, the double GL-transformation, does.

An *alternating fixpoint* of program $\mathcal{P}$ is an interpretation I of $\mathcal{P}$ such that

$$I = \mathbf{S}_P(\mathbf{S}_P(I))$$

Let $\mathbf{A}_\mathcal{P}(I)$ be $\mathbf{S}_P(\mathbf{S}_P(I))$. By [2, 1], $\mathbf{S}_P$ is antimonotonic, that is, $\mathbf{S}_P(I) \subseteq \mathbf{S}_P(J)$ if $J \subseteq I$, and $\mathbf{A}_\mathcal{P}$ is monotonic, whose least fixpoint is given by $\mathbf{A}_\mathcal{P} \uparrow \omega$. Therefore, every logic program has at least one alternating fixpoint.

Proposition 2.2 ([2]) Let T be $\mathbf{A}_\mathcal{P} \uparrow \omega$, the least fixpoint of $\mathbf{A}_\mathcal{P}$, $F = \{a | a \notin \mathbf{S}_P(T)\}$. Then $< T, F >$ is the well-founded model of $\mathcal{P}$. $\Box$

In fact, the well-founded semantics can also be characterized by the set of all alternating fixpoints. The proof of the following corollary follows from the above proposition and the fact that $\mathbf{S}_{\mathcal{P}}$ is antimonotonic and $\mathbf{A}_{\mathcal{P}}$ is monotonic.

Corollary 2.3 A literal L is true in the well-founded semantics of $\mathcal{P}$ if and only if it is true in every alternating fixpoint of $\mathcal{P}$. $\Box$

Baral and Subrahmanian proposed the stable class semantics to resolve the no-stable-model problem [1]. A stable class of program $\mathcal{P}$ is a set S of interpretations of $\mathcal{P}$ such that

$$S = \{\mathbf{S}_P(I) | I \in S\}$$

Then the sable class semantics is defined by the union of all minimal strict stable classes of $\mathcal{P}$, based on the preference relation defined in [1]. They have also shown that the stable and the stable class semantics coincide for stratified programs.

3 Problems with Existing Semantics

All three prominent semantics can be characterized by the fixpoints of GL-based transformations. However the following examples demonstrate various problems with these three semantics which must be addressed.

Example 3.1 Consider $\mathcal{P}$ given by

$$\begin{aligned} a &\leftarrow \neg b \\ b &\leftarrow \neg a \\ p &\leftarrow \neg a \\ p &\leftarrow \neg p \end{aligned}$$

This program has a unique stable model $\{b, p\}$. Due to the self-recursion of the fourth clause, it is very difficult to determine the value of p. However such difficulty shall not be used to justify the truth values of both b and p. $\Box$

The above example shows that, other than the inconsistency problem, the stable semantics also suffers from the biased-truth-assignment problem. The well-founded semantics, on the other hand, may not be adequate to characterize the meaning of logic programs, as demonstrated by the following example.

Example 3.2 ([1]) Consider the following program $\mathcal{P}$

$$\begin{aligned} a &\leftarrow \neg b \\ b &\leftarrow \neg a \\ c &\leftarrow a \\ c &\leftarrow b \end{aligned}$$

$\mathcal{P}$ has four alternating fixpoints, that is, $m_1 = \{a, c\}$ and $m_2 = \{b, c\}$, $m_3 = \emptyset$, and $m_4 = \{a, b, c\}$, Among which, the first two are stable models.

Our intuition tells us that c should be true according to the semantics of P, while both a and b should have an unknown truth value. These are exactly the truth values assigned by the stable semantics. However, the well-founded semantics assigns the unknown to all three atoms, due to the very existence of two extra alternating fixpoints that are not fixpoints, namely m_3 and m_4. $\square$

The stable class semantics characterizes the intuitive meaning for programs in the above two examples. However, the following example clearly shows that the stable class semantics may yield unreasonable conclusions.

Example 3.3 Let $\mathcal{P}$ be given by

$$\begin{aligned} &a \leftarrow \neg a \\ &b \leftarrow \neg b \\ &c \leftarrow a, \neg a \\ &c \leftarrow b, \neg b \end{aligned}$$

$\mathcal{P}$ has two strict stable classes, viz. $C_1 = \{\{a, b, c\}, \emptyset\}$, and $C_2 = \{\{a, c\}, \{b, c\}\}$, but only C_2 is minimal. Therefore, the stable class semantics of $\mathcal{P}$ is determined by C_2, which implied c is true.

Since the premises for c can not be satisfied in any circumstance, c shall not be true in any reasonable semantics. $\square$

4 Minimal Well-Founded Semantics

The well-founded semantics is based on the alternating fixpoints and therefore, avoids the no-fixpoint problem. However, not every alternating fixpoint makes positive contributions to defining semantics. We believe the inadequateness of the well-founded semantics is due to those undesirable alternating fixpoints such as m_3 and m_4 in Example 3.2.

The stable class semantics uses the preference relation between the stable classes to rule out those undesirable stable classes. However, the preference relation used by Baral and Subrahmanian is solely based on the *largeness* of stable classes, which may retain wrong stable classes.

Consider $\mathcal{P}$ in Example 3.3 again. $\mathcal{P}$ has two strict stable classes C_1 and C_2 that are comprised of four alternating fixpoints of $\mathcal{P}$, viz. $m_1 = \{a, b, c\}$, $m_2 = \emptyset$, $m_3 = \{a, c\}$, and $m_4 = \{b, c\}$. The undesirable conclusion of the stable class semantics may be avoided if C_1 is preferred to C_2, not vice versa. Readers may notice that among all four alternating fixpoints of $\mathcal{P}$, m_1 is the only one that is also a model of $\mathcal{P}$.

We take a different approach to eliminating undesirable alternating fixpoints.

First, we believe only those alternating fixpoints that are models of program $\mathcal{P}$ are of our interest. An alternating fixpoint which is not a model of the program may yield conclusions based on inconsistent assumptions. Secondly, some alternating fixpoints are larger than others and therefore should be eliminated, as otherwise, little negative interpretations may be deduced.

In the following definition we first define *minimal alternating fixpoints* of programs to eliminate those undesirable alternating fixpoints, and then define the *minimal well-founded semantics* based on the minimal alternating fixpoints.

Definition 4.1 Let $\mathcal{P}$ be a program and I an alternating fixpoint of $\mathcal{P}$. Then I is said to be

1. an *m-alternating fixpoint* [1] of $\mathcal{P}$ if I is a model of $\mathcal{P}$;
2. a *minimal alternating fixpoint* of $\mathcal{P}$ if
   (a) I is an m-alternating fixpoint of $\mathcal{P}$, and
   (b) no proper subset of I is an m-alternating fixpoint of $\mathcal{P}$; and
3. a *minimal well-founded model* of $\mathcal{P}$ if there exists a minimal alternating fixpoint J of $\mathcal{P}$ such that either $I = J$ or $I = \mathbf{S}_{P}(J)$.

The *minimal well-founded semantics* of $\mathcal{P}$ is then defined by the set of all minimal well-founded models of $\mathcal{P}$. $\square$

Example 4.1 $\mathcal{P}$ in Example 3.1 has five alternating fixpoint, viz.

$$m_1 = \{b, p\},\ m_2 = \{a, p\},\ m_3 = \{a\},\ m_4 = \{a, b, p\},\ m_5 = \emptyset.$$

Of which, m_1 and m_2 are only minimal alternating fixpoints, and $m_3 = \mathbf{S}_P(m_2)$. Therefore, the minimal well-founded models are m_1, m_2, and m_3.

The stable semantics is biased toward b and p, while the minimal well-founded semantics as well as the stable class semantics characterize the intuitive meaning of $\mathcal{P}$. $\square$

Example 4.2 The set of all minimal well-founded models of $\mathcal{P}$ in Example 3.2 is m_1 and m_2, i.e., the set of all its stable models. $\square$

[1]According to Lemma 4.1, an m-alternating fixpoint can also be defined as an alternating fixpoint I such that $\mathbf{S}_P(I) \subseteq I$.

Example 4.3 Consider $\mathcal{P}$ in Example 3.3 again. $\mathcal{P}$ has only one minimal alternating fixpoint, viz. $m_1 = \{a, b, c\}$, and $m_2 = \mathbf{S}_P(m_1) = \emptyset$. Therefore, the minimal well-founded semantics is determined by m_1 and m_2. $\Box$

In summary, among all four semantics discussed so far, the minimal well-founded semantics is the only one that adequately characterizes the intuitive meaning of all logic programs discussed in the previous section. Furthermore, the following theorem shows that the minimal well-founded semantics is always consistent.

First, we show a utility lemma.

Lemma 4.1 An alternating fixpoint I of $\mathcal{P}$ is a model of $\mathcal{P}$ if and only if $\mathbf{S}_P(I) \subseteq I$.
Proof: ($\Rightarrow$) Assume I is a model of $\mathcal{P}$. Then I is also a model of $\mathcal{P}^I$. Since $\mathbf{S}_P(I)$ is the least model of $\mathcal{P}^I$, $\mathbf{S}_P(I) \subseteq I$.

($\Leftarrow$) Assume $\mathbf{S}_P(I) \subseteq I$ and there exists a clause

$$a \leftarrow b_1, \ldots, b_n, \neg c_1, \ldots, \neg c_m$$

in $\mathcal{P}$ that is false in I, that is, b_i's are in I, c_j's are not in I, and a is not in I. Since $\mathbf{S}_P(I) \subseteq I$ and $\mathbf{S}_P(\mathbf{S}_P(I)) = I$, c_j's are not in $\mathbf{S}_P(I)$ and b_i's are in $\mathbf{S}_P(\mathbf{S}_P(I))$, which implies a is contained in $\mathbf{S}_P(\mathbf{S}_P(I))$. This contradicts to the fact that $a \notin I$ and $I = \mathbf{S}_P(\mathbf{S}_P(I))$. $\Box$

Theorem 4.1 Any logic program has at least one minimal well-founded model.
Proof: It is sufficient to construct an alternating fixpoint that is also a model of $\mathcal{P}$.

Let $I = \mathbf{A}_{\mathcal{P}} \uparrow \omega$, i.e., the least fixpoint of $\mathbf{A}_{\mathcal{P}}$, and $J = \mathbf{S}_P(I)$. Then J is also an alternating fixpoint of $\mathcal{P}$, since $I = \mathbf{S}_P(\mathbf{S}_P(I)) = \mathbf{S}_P(J)$.

Since I is the least alternating fixpoint, we have $I \subseteq J$, i.e., $\mathbf{S}_P(I) \subseteq I$. By Lemma 4.1, I is a model of $\mathcal{P}$. $\Box$

5 Relationship with the Stable Class Semantics

In this section, we compare the minimal well-founded semantics with the stable class semantics.

Let $\mathcal{P}$ be a program and S a stable class of $\mathcal{P}$. The *lowest upper bound* of S, denoted as $lub(S)$, is the smallest interpretation I such that every interpretation in S is a subset of I. A stable class is said to be *normal* if $lub(S) \in S$. Obviously, not every stable class is normal. For example, C_1 in Example 3.3 is normal but C_2 is not. The meaning of a normal stable class is well-defined since the class contains its lowest upper bound. In our approach, we require that the concerned alternating fixpoints be models of the program. In the following theorem we are gong to show that there exists a one-to-one correspondence between the m-alternating fixpoint and the normal stable class.

Theorem 5.1 I is an m-alternating fixpoint of program $\mathcal{P}$ if and only if there exists a normal stable class S of $\mathcal{P}$ such that $I = lub(S)$.

Proof: ($\Rightarrow$) Assume I is an m-alternating fixpoint of $\mathcal{P}$. Let $J = \mathbf{S}_P(I)$. Then $\{I, J\}$ is a stable class of $\mathcal{P}$. Since I is a model of $\mathcal{P}$, by Lemma 4.1, $J \subseteq I$, and therefore, $\{I, J\}$ is normal and $I = lub(\{I, J\})$.

($\Leftarrow$) Assume S is a normal stable class of $\mathcal{P}$ and $I = lub(S)$. Let $J = \mathbf{S}_P(I)$ and $K = \mathbf{S}_P(J) = \mathbf{S}_P(\mathbf{S}_P(I)) = \mathbf{A}_{\mathcal{P}}(I)$. Since S is a stable class, $J \in S$ and $K \in S$, which implies that $K \subseteq I$. Furthermore, since $I \in S$ and S is a stable class, there exist I_1 and I_2 in S such that $I = \mathbf{S}_P(I_1)$ and $I_1 = \mathbf{S}_P(I_2)$, that is, $I = \mathbf{A}_{\mathcal{P}}(I_2)$ and $I_2 \in S$. It follows that $I \subseteq K$ since $I_2 \subseteq I$ and $\mathbf{A}_{\mathcal{P}}$ is monotonic. Therefore, $I = K$, since $K \subseteq I$. It follows that I is an alternating fixpoint. Furthermore, since $\mathbf{S}_P(I) \subseteq I$, by Lemma 4.1, I is an m-alternating fixpoint of $\mathcal{P}$. $\Box$

From the above proof we can see that each normal stable class S contains a normal class S' such that (1) S' has at most two interpretations and (2) S and S' have the same lowest upper bound. Furtheremore, Let $I = lub(S)$ for some normal stable class S. Since $\mathbf{S}_P$ is antimonotonic, $\mathbf{S}_P(I)$ is the greatest lower bound of S and $\mathbf{S}_P(I)$ is in S. Therefore, the meaning of a normal stable class is always determined by its lowest upper bound and greatest lower bound, not any other interpretations.

A normal stable class S of $\mathcal{P}$ is said to be *minimal* if there exists no normal stable class S' of $\mathcal{P}$ such that $lub(S') \subset lub(S)$. Then, by Theorem 5.1, the minimal well-founded semantics is determined by the set of all minimal normal stable classes.

Corollary 5.2 The minimal well-founded semantics is determined by the union of all minimal normal stable classes. $\Box$

6 Autoepistemic Circumscription

In this section, we demonstrate various equivalences between the semantics of logic programs and the autoepistemic circumscription theories.

Circumscription has been proposed to express the idea that the extensions of abnormal predicates should be minimized. Let $T(Q, P)$ be a logic theory, where Q, P are disjoint predicates in T. Then $CIRC(T(Q, P); P)$ is used to denote the circumscription of T on P, i.e., a second order theory in which the extension of P has been minimized [8, 9, 7].

Gelfond has first proposed to represent a logic program with negation as an autoepistemic theory by replacing negative literals $\neg c$ with $\neg \mathcal{L}c$, standing for *believing in* $\neg c$ [4].

Definition 6.1 Assume $\mathcal{P}$ is a logic program with a set P of predicates. Let $\mathcal{L}P$ be the set of belief predicates whose negation standing for "*believing in* $\neg P$". Then $\mathcal{P}(P, \mathcal{L}P)$ is used to represent the corresponding autoepistemic theory (also called the belief theory) consisting of clauses

$$a \leftarrow b_1 \wedge \cdots \wedge b_n \wedge \neg\mathcal{L}c_1 \wedge \cdots \wedge \neg\mathcal{L}c_m.$$

$\square$

Example 6.1 $\mathcal{P} = \{a \leftarrow \neg b;\quad p \leftarrow a, \neg q;\quad q \leftarrow b, \neg p\}$ can be represented by a belief theory $\mathcal{P}(P, \mathcal{L}P) = \{a \leftarrow \neg\mathcal{L}b;\quad p \leftarrow a, \neg\mathcal{L}q;\quad q \leftarrow b, \neg\mathcal{L}p\}$. Note that $P = \{a, b, p, q\}$ and $\mathcal{L}P = \{\mathcal{L}a, \mathcal{L}b, \mathcal{L}p, \mathcal{L}q\}$. $\square$

For convenience, from now on, we identify a program $\mathcal{P}(P)$ with its corresponding autoepistemic theory $\mathcal{P}(P, \mathcal{L}P)$ if there is no confusion.

More notations here. Assume $T(P, \mathcal{L}P)$ is a logic theory. A P-interpretation (or $\mathcal{L}P$-interpretation) of T is an interpretation of T containing only atoms whose predicates are from P (or $\mathcal{L}P$). A *P-model* I of T is a P-interpretation such that there exists an $\mathcal{L}P$-interpretation J and $I \cup J$ is a model of T. An $\mathcal{L}P$-model is defined similarly. Suppose $F(P)$ is a logic theory, then $F(\mathcal{L}P)$ is the theory obtained from $F(P)$ by replacing each predicate p with $\mathcal{L}p$.

Theorem 6.1 Let $\mathcal{P}(P, \mathcal{L}P)$ be a logic program and I be a P-interpretation of $\mathcal{P}$. Then

1. $\mathbf{S}_P(I) = \{a | CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge T_{I_\mathcal{L}} \models a\}$, where $T_{I_\mathcal{L}} = \{\neg\mathcal{L}a | a \notin I\}$.

2. I is a fixpoint of $\mathcal{P}$ if and only if I is a P-model of

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge (P \equiv \mathcal{L}P)$$

   where $P \equiv \mathcal{L}P$ means $a \leftrightarrow \mathcal{L}a$ for each atom $a \in P$.

3. I is an alternating fixpoint of $\mathcal{P}$ if and only if I is a P-model of

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge CIRC(\mathcal{P}(\mathcal{L}P, P); \mathcal{L}P)$$

Proof: (1) a is contained in $\mathbf{S}_P(I)$ if and only if $\mathcal{P}^I \models a$, that is, if and only if

$$CIRC(\mathcal{P}^I(P); P) \models a$$

However, $CIRC(\mathcal{P}^I(P); P)$ and $CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge T_{I_\mathcal{L}}$ is equivalent as far as the P-interpretation is concerned. (2) and (3) follow from (1). $\square$

Then both the stable and the well-founded semantics can be characterized by autoepistemic circumscription theories. The proof of the following theorem is straightforward.

Theorem 6.2 1. The stable semantics of program $\mathcal{P}$ is characterized by

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge (P \equiv \mathcal{L}P)$$

2. The well-founded semantics of program $\mathcal{P}$ is characterized by

$$CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge CIRC(\mathcal{P}(\mathcal{L}P, P); \mathcal{L}P)$$

□

Now we define the minimal well-founded semantics in terms of autoepistemic circumscription. Suppose $\mathcal{P}(P, \mathcal{L}P)$ is a logic program, then $T_{wf}(P, \mathcal{L}P)$ is used to denote

$$\mathcal{P}(\mathcal{L}P) \wedge CIRC(\mathcal{P}(P, \mathcal{L}P); P) \wedge CIRC(\mathcal{P}(\mathcal{L}P, P); \mathcal{L}P)$$

Theorem 6.3 A formula $F(P)$ is true in the minimum well-founded semantics of $\mathcal{P}$ if and only if $F(P) \wedge F(\mathcal{L}P)$ is a logical consequence of

$$T_{wf}(P, \mathcal{L}P) \wedge \neg\exists(P', \mathcal{L}P')(T_{wf}(P', \mathcal{L}P') \wedge (\mathcal{L}P' \subset \mathcal{L}P))$$

where $\mathcal{L}P' \subset \mathcal{L}P$ mean the extension of $\mathcal{L}P'$ is the proper subset of that of $\mathcal{L}P$.
Proof: First, we specify some notations. Assume I is a P-interepretation, then $I_{\mathcal{L}}$ denotes the corresponding $\mathcal{L}P$-interpretation, that is, $I_{\mathcal{L}} = \{\mathcal{L}a \mid a \in I\}$.

Let $T_{mwf}(P, \mathcal{L}P)$ denote

$$T_{wf}(P, \mathcal{L}P) \wedge \neg\exists(P', \mathcal{L}P')(T_{wf}(P', \mathcal{L}P') \wedge (\mathcal{L}P' \subset \mathcal{L}P)).$$

By Theorem 6.1 (3), I is an m-alternating fixpoint of program $\mathcal{P}$ if and only if $I_{\mathcal{L}}$ is an $\mathcal{L}P$-model of $T_{wf}(P, \mathcal{L}P)$. Therefore, I is a minimal alternating fixpoint of $\mathcal{P}$ if and only if $I_{\mathcal{L}}$ is an $\mathcal{L}P$-model of $T_{mwf}(P, \mathcal{L}P)$.

Assume I and J are two P-interpretations such that $I \cup J_{\mathcal{L}}$ is a model of $T_{wf}(P, \mathcal{L}P)$. Then by Theorem 6.1 (3), $I = \mathbf{S}_P(J)$ and $J = \mathbf{S}_P(I)$. Therefore, I is a minimal well-founded model of $\mathcal{P}$ if and only if either I is a P-model of T_{mwf} or $I_{\mathcal{L}}$ is an $\mathcal{L}P$-model of T_{mwf}. It follows that $F(P)$ is true in the minimal well-founded semantics of $\mathcal{P}$ if and only if $F(P)$ is true in every P-model of T_{mwf} and $F(\mathcal{L}P)$ is true in every $\mathcal{L}P$-model of T_{mwf}.
□

7 Conclusions

We propose the minimal well-founded semantics based on the alternating fixpoints of programs, and demonstrate that the minimal well-founded semantics has overcome many existing problems associated with the stable, the well-founded, and the stable class semantics. We also demonstrate various equivalences between the semantics of logic programs and the autoepistemic circumscription formulas.

Our work is inspired by that of Baral and Subrahmanian. However, two approaches are quite different. By considering all stable classes, the stable class semantics suffers from the problem that unreasonable conclusions may be deduced. On the other hand, we consider only those alternating fixpoints which are also models of the program and therefore, characterize the intuitive meanings of logic programs.

We have demonstrated that the stable class semantics would be the same as the minimal well-founded semantics if only normal stable classes are considered, and therefore, the unreasonable conclusions can be avoided.

Baral and Subrahmanian have extended the stable class semantics into the default theory [1]. We believe our approach can also be used to define the extended semantics for the default theory which will be discussed in our forthcoming paper.

References

[1] C.R. Baral and V.S. Subrahmanian. Stable and extension class theory for logic programs and default logics. *Journal of Automated Reasoning*, 345 – 366, 1992.

[2] A. Van Gelder. The alternatin fixpoints of logic programs with negation. In *Proceedings of the 8th ACM PODS*, pages 1 – 10, 1989.

[3] A. Van Gelder, K. Ross, and J.S. Schlipf. The well-founded semantics for general logic programs. *JACM*, 38:620 – 650, 1991.

[4] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of AAAI-87*, 1987.

[5] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of the 5th Intl. Conference and Symposium on Logic Programming*, pages 1070–1080, 1988.

[6] Y. Hu and L. Yuan. Extended well founded semantics for logic programs with negations. In *Proceedings of the 8th International Conference on Logic Programming*, pages 412 – 425, 1991.

[7] V. Lifschitz. Computing circumscription. In *Proceedings of the 9th Int. Joint Conference on AI*, 1986.

[8] J. McCarthy. Applications of circumscription to formalizing common sense knowledge. *AI*, 28:89–116, 1986.

[9] J. McCarthy. Circumscription – a form of non-monotonic reasoning. *AI*, 13:27–39, 1980.

[10] T.C. Przymusinski. Autoepistemic logics of closed world beliefs and logic programming. In *Proc. of the Workshop on Nonmonotonic Reasoning and Logic Programming*, pages 3–20, 1991.

[11] L. Yuan and J. You. Autoepistemic circumscription and logic programs. *Journal of Automated Reasoning*, to appear.