Pakistan Journal of Applied Sciences 2(10): 980-984, 2002

Extended Stable Models for Logical Programs with Many Negations

Victor Felea
Faculty of Computer Science, University "Al. I. Cuza" IASI
6600 IASI, Romania

Abstract: The family of stable models for a logic program with one negation was studied by Melvin Fitting. We introduce extended stable model semantics of logic programs with many negations, which natural extends the notion of stable model semantics for logic programs with one negation. We use the notion of bilattice with two orderings which defines the structure of the family of stable models. The first one is called knowledge ordering, the second one is called degree of truth. For a vector of valuations in a bilattice $\boldsymbol{B}$, we define a pseudovaluation and an operator associated to a program $\wp$. We also consider the notion of i-model for a program $\wp$. For an operator we define a fixed-point iteration. This iteration is applied to the operator associated to $\wp$ and produces so-called extended stability operators. When the fixed-point iteration can be applied by n times, where n is the number of negations, then every fixed point of the last operator is an extended stable valuation of $\wp$.

Key Words: Logic Program, Model, Fixed Point, Stable Model, Valuation, Pseudovaluation, Stability Operator

Introduction

Gelfond and Lifschitz (1988) have extended general logic programs (with negation-as-failure interpreted autoepistemically) by adding "classical" negation which is, in fact, a strong negation (in sense of Gerd Wagner, 1991).

Kowalski and Sadri (1990), also Przymusinski (1989) consider logic programs with two kinds of negations.

Da Costa *et al.*, (1990) introduce a resolution calculus for logics with weak and strong negation based on idea of paraconsistency.

Gerd Wagner (1991) use the concept of strong negation for representation and processing of explicit negative information and the concept of weak negation for the usual form of negation in databases, viz.negation-as-failure.

Melvin Fitting (1993 and 1991) studies the structure of the family of all stable models for logic programs and uses bilattices with two orderings: one is knowledge ordering, denoted by $\leq_k$ and another is degree of truth, which is denoted by $\leq_t$.

Rajasekar and Jack Minker defined in (1989) a model-theoretic semantics based on states for stratified disjunctive programs and shown that the least state characterized by the fix-point semantics is a stable-state and a supported state.

Kifer (1998) considered several decidability problems of a query predicate with respect to the least fix-point models of a logic program $\wp$, and with respect to any fixpoint model of $\wp$.

Spyratos and Stamate (1996) use the logic $L(2)$ with five values (true, false, unknown, possible true, possible false) and define logic programs which may contain logical values in the body of clauses. They study the stable multivalue semantics for programs with a negation.

Preliminaries: We use the notion of bilattice with two inter-related orderings, denoted by $\leq_t$, $\leq_k$. The first one is on the "degree of truth", the second one is the "degree of knowledge".

Definition 1: A pre-bilattice is a structure $< B, \leq_t, \leq_k >$, where $\boldsymbol{B}$ is a nonempty set, $\leq_t$ and $\leq_k$ are each partial orderings giving $\boldsymbol{B}$ the structure of a lattice with top and bottom.

Definition 2: In a pre-billatice $< B, \leq_t, \leq_k >$, meet and join under $\leq_t$ are denoted by $\wedge$ and $\vee$, meet and join under $\leq_k$ are denoted by $\otimes$ *and* $\oplus$ respectively. True and false are top and bottom under $\leq_t$, top and bottom under $\leq_k$ are denoted - and $\perp$. If the pre-bilattice is complete, infinitary meet and join under $\leq_t$ are denoted $\bigwedge$ and $\bigvee$, and infinitary meet and join under $\leq_k$ are denoted Π and Σ.

Definition 3: A distributive bilattice is a pre-bilattice $< B, \leq_t, \leq_k >$ in which all 12 distributive laws involving $\wedge, \vee$, $\otimes$ and $\oplus$ hold. An infinitely distributive bilattice is a complete pre-bilattice in which all infinitary and finitary distributive laws hold.

Definition 4: A bilattice has a pseudonegation if there is a mapping $\rceil$ that reverses the $\leq_t$ ordering and leaves unchanged the $\leq_k$ ordering. A negation is a pseudonegation with the property $\rceil\rceil x = x$. For other notions and results (Melvin Fitting, 1993 and 1991).

Logic Programs with Many Pseudonegations: Let B be a complete bilattice with n pseudonegations, denoted by $\sim_i$, $i = \overline{1,n}$. Assume that $\wedge$ and $\vee$ are duals with respect to each $\sim_i$ and also $\exists$ and $\forall$. It results that the pseudonegations can be pushed all the way inside. We treat members of B as atoms and allow them to appear in clause bodies. A formula that does not contain a member of B is called a pure formula.

We take a clause to have the form $A \leftarrow \alpha$, where A is the head (an atomic formula) and α is a formula (the body of the clause).

Literals have the form A or $\rceil_{\alpha_1} \rceil_{\alpha_2} \ldots \rceil_{\alpha_k} A$, where A is an atomic formula and $\rceil_{\alpha_j}$ are pseudonegations, $j = \overline{1,k}$.
A formula is built up from literals and elements of B using the operators $\wedge, \vee, \exists, \rceil_{\alpha_j}$. We assume that all free variables of the clause body appear in the head, the head contain no function symbol, and a given predicate letter can occur in the head of only a single clause, by the combining all the clauses having as the head an atomic formula with the same one function symbol into a single clause (Fitting Melvin, 1993).
Therefore a clause has the form:

$P(x_1, \ldots, x_n) \leftarrow \varphi(x_1, \ldots, x_n)$

where $P(x_1, \ldots, x_n)$ is the head of formula and $\varphi(x_1, \ldots, x_n)$ is the body of formula.
A logical program is a finite set of clauses with no predicate letter appearing in the head of more than one clause. For a program $\wp$, we denote by $\wp^*$ the set of all ground instances of members of $\wp$ over Herbrand base. Let B be a bilattice.
A valuation in B is a mapping v from pure ground atoms to members of B. The family of all valuations in B is denoted $\gamma(B)$. It is possible to define in $\gamma(B)$ two pointwise orderings:

(1) $v_1 \leq_t v_2$ if and only if $v_1(A) \leq_t v_2(A)$ for every pure ground atom A;

(2) $v_1 \leq_k v_2$ if and only if $v_1(A) \leq_k v_2(A)$ for every pure ground atom A.

The set of valuations in B itself is a billatice and inherits the structure of B (Fitting Melvin, 1993).
A valuation is natural extended from pure ground atoms to all formulas. If $c \in B$, then $v(c) = c$ and

$v(\rceil_{\alpha_1} \rceil_{\alpha_2} \ldots \rceil_{\alpha_k} A) = \sim_{\alpha_1} \sim_{\alpha_2} \ldots \sim_{\alpha_k} v(A)$.

Moreover, $v(f_1 \vee f_2) = v(f_1) \vee v(f_2)$, where $\vee$ on the right is the join of B in the $\leq_t$ ordering.
Similarly for $v(f_1 \wedge f_2)$.
$v((\exists x)\varphi(x)) = \sum v(\varphi(t))$, for all closed term t.
$v((\forall x)\varphi(x)) = \prod v(\varphi(t))$, for all closed term t.
Here $\prod$ and $\sum$ are infinitary meet and join under $\leq_k$.

Generalized Transformation of Type Gelfond-Lifschitz: We use the idea of Melvin Fitting (1993) of separating positive information and negative information, by defining of consequence operator, which will reflect this separation. This consequence operator will accept as input a vector of valuations $\bar{v} = (v_1, \ldots, v_{n+1})$, where v_1 is for assigning meanings to positive literals, and v_i is for pseudonegation $\rceil_{i-1}$, $2 \leq i \leq n+1$.

For a vector of valuations $\bar{v} = (v_1, \ldots, v_{n+1})$, we define a pseudovaluation denoted $v_1 \Delta v_2 \Delta \ldots \Delta v_{n+1}$.

Definition 5: Let $\bar{v} = (v_1, \ldots, v_{n+1})$ a vector of valuations in the bilattice B with n negations $\sim_i$, $i = \overline{1,n}$. A pseudovaluation in B, denoted $v_1 \Delta v_2 \Delta \ldots \Delta v_{n+1}$, is defined as follows:

$(v_1 \Delta v_2 \Delta \ldots \Delta v_{n+1})(A) = v_1(A)$

$(v_1 \Delta v_2 \Delta \ldots \Delta v_{n+1})(\rceil_i A) = \sim_i v_{i+1}(A)$, $i = \overline{1,n}$

$(v_1 \Delta v_2 \Delta \ldots \Delta v_{n+1})(\rceil_{\alpha_1} \ldots \rceil_{\alpha_k} A) = \sim_{\alpha_1} \sim_{\alpha_2} \ldots \sim_{\alpha_k} v_{\alpha_k+1}(A)$

$k \geq 2$; $\alpha_j \in \{1, 2, \ldots, n\}$, $j = \overline{1,k}$.

These pseudovaluations are extended to formulas by induction.
We use the term B-pseudovaluation (shortly pseudovaluation) for a mapping from pure ground literals to B.
For a pseudovaluation v and a literal $\rceil_{\alpha_1} \ldots \rceil_{\alpha_k} A$, the value assigned to $\rceil_{\alpha_1} \ldots \rceil_{\alpha_k} A$ is generally different from the value $\sim_{\alpha_1} \sim_{\alpha_2} \ldots \sim_{\alpha_k} v(A)$.
Now we define an extended immediate consequence operator $\Psi_\wp$ for logic programs with many negations.

Definition 6 : The extended immediate consequence perator $\Psi_\wp : \gamma(B)^{n+1} \to B$ is defined as follows. Let $(v_1, \ldots, v_{n+1})$ be a vector of n + 1-valuations in B. $\Psi_\wp(v_1, \ldots, v_{n+1})$ is the valuation such that the following conditions hold:

(1) $\Psi_\wp(v_1, \ldots, v_{n+1})(A)$ = false, if the pure ground atom A is not the head of any member of $\wp^*$.

(2) If $A \leftarrow \alpha$ is in $\wp^*$, then $\Psi_\wp(v_1, \ldots, v_{n+1})(A) = (v_1 \Delta \ldots \Delta v_{n+1})(\alpha)$.

For n = 1 it obtains the operator $\Psi_\wp$ defined by Fitting (1993).
In the following we need the notion i-model for a program $\wp$.

Definition 7: A vector of n + 1 valuations $\bar{v} = (v_1, \ldots, v_{n+1})$ is an i-model for the program $\wp$ provided the following conditions are fulfilled:

(1) $v_i(A)$ = false if there is no member of $\wp^*$ of the form $A \leftarrow \alpha$.

(2) $v_i(A) = (v_1 \Delta \ldots \Delta v_{n+1})(\alpha)$ if $A \leftarrow \alpha \in \wp^*$.

Remark 1: The vector $\bar{v} = (v_1, \ldots, v_{n+1})$ of valuations is an i-model for $\wp$ if and only if $\Psi_\wp(v_1, \ldots, v_{n+1}) = v_i$, that is v_i is an i-fixed point of $\Psi_\wp$.
In the following several notations are necessary.
For a vector of n + 1 valuations, $\bar{v} = (v_1, \ldots, v_{n+1})$, let us denote by $\overline{v_i}$ the vector $(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n+1})$.
That means $\overline{v_i}$ has all the components of $\bar{v}$

with exception of v_j. Let $f = \Psi_{\wp}$ and $g_1(\bar{v}_h)$ be an h-fixed point for f.

Let $\bar{v}_{h,i} = (v_1, ..., v_{h-1}, v_{h+1}, ..., v_{i-1}, v_{i+1}, ..., v_{n+1})$. Assume that $h < i$.

Let $g_2(\bar{v}_{h,i})$ be an i-fixed point for $g_1(\bar{v}_h)$.

Proposition 1: We have:

(a) $g_2(\bar{v}_{h,i})$ is an i-fixed point for f $(v_1, ..., g_2(\bar{v}_{h,i}), ..., v_i, ..., v_{n+1})$, where the argument $g_2(\bar{v}_{h,i})$ of f is situated on the position h.

(b) The vector $(v_1, ..., g_2(\bar{v}_{h,i}), v_{h+1}, ..., v_{i-1}, g_2(\bar{v}_{h,i}), ..., v_{n+1})$, where $g_2(\bar{v}_{h,i})$ is in the positions h and i, is at the same time h-model and i-model for $\wp$.

Proof: Since $g_1(\bar{v}_h)$ is an h-fixed point for f, it follows

$$g_1(\bar{v}_h)=f(v_1,...,v_{h-1},g_1(\bar{v}_h),v_{h+1},...,v_{n+1}) \quad (4.1)$$

Because $g_2(\bar{v}_{h,i})$ is an i-fixed point for $g_1(\bar{v}_h)$ we have

$$g_2(\bar{v}_{h,i})=g_1(v_1, ..., v_{h-1}, v_{h+1}, ..., v_{i-1}, g_2(\bar{v}_{h,i}),v_{i+1}, ..., v_{n+1}) \quad (4.2)$$

Assume that $h < i$.

If we take $g_2(\bar{v}_{h,i})$ instead of v_i in the relation (4.1) we obtain:

$$G_2=f(v_1, ..., v_{h-1}, G_2, v_{h+1}, ..., v_{i-1}, g_2(\bar{v}_{h,i}), v_{i+1}, ..., v_{n+1}) \quad (4.3)$$

where

$$G_2 = g_1(v_1, ..., v_{h-1}, v_{h+1}, ..., v_{i-1}, g_2(\bar{v}_{h,i}), v_{i+1}, ..., v_{n+1}).$$

By the relation (4.2), we have $G_2 = g_2(\bar{v}_{h,i})$, therefore it follows:

$$g_2(\bar{v}_{h,i}) = f(v_1, ..., v_{h-1}, g_2(\bar{v}_{h,i}), v_{h+1}, ..., v_{i-1}, g_2(\bar{v}_{h,i}),v_{i+1}, ..., v_{n+1}) \quad (4.4)$$

The statement (b) results from (a).

The following proposition considers fixed points for f, $g_1(\bar{v}_h)$ and $g_2(\bar{v}_{h,i})$.

Proposition 2: Let $g_1(\bar{v}_h)$, $g_2(\bar{v}_{h,i})$ as in the proposition 1. Let $1 \le h < i < j \le n+1$ and $g_3(\bar{v}_{h,i,j})$ be a j-fixed point for $g_2(\bar{v}_{h,i})$. We have:

a) $g_3(\bar{v}_{h,i,j})$ is a j-fixed point for f $(v_1, ..., v_{h-1}, g_3(\bar{v}_{h,i,j}), ..., v_{i-1}, g_3(\bar{v}_{h,i,j}), ..., v_j, ..., v_{n+1})$.

b) The vector $(v_1, ..., v_{h-1}, v', ..., v_{i-1}, v', ..., v_{j-1}, v', v_{j+1}, ..., v_{n+1})$ is at the same time an h-model, i-model and j-model for $\wp$, where $v' = g_3(\bar{v}_{h,i,j})$ and $\bar{v}_{h,i,j}$ is the vector which has all components of $\bar{v}_{h,i}$ with except of v_j.

The proof is similar with that of proposition 1.

For these fixed points we give the general case of this iteration. Let us consider some notations.

Notations 1: Let β be the vector of integers (1, 2, ..., n + 1). Let β_k be a subset of k elements from β. Let γ_k be the difference between β and β_k : $\gamma_k = \beta - \beta_k$. If $\gamma_k = (i_1, ..., i_p)$, where p = n + 1 - k, then we denote $\bar{v}_{\gamma_k} = (v_{i_1}, ..., v_{i_p})$.

Theorem 1: Let $g_{h+1}(\bar{v}_{\gamma_{h+1}})$ be a φ_h-fixed point of $g_h(\bar{v}_{\gamma_h})$ for every h, $0 \le h \le k$, where $0 < k < n + 1$, $\varphi_h \in \gamma_h$, $\gamma_0 = \beta$; $g_0=f= \Psi_{\wp}$, $\beta_0= \emptyset$, $\gamma_{h+1} = \gamma_h - \{\varphi_h\}$, $\beta_{h+1} = \beta - \gamma_{h+1}$ for each h, $0 \le h \le k$. The following statements hold:

a) $g_{k+1}(\bar{v}_{\gamma_{k+1}})$ is a φ_k-fixed point for $f(\acute{W}_1, ..., \acute{W}_{n+1})$, where $w'_j = v_j$ for each $j \in \gamma_k$ and $w'_j = g_{k+1}(\bar{v}_{\gamma_{k+1}})$ for each $j \in \beta_k$.

b) The vector $(w''_1, ..., w''_{n+1})$ is a j-model for the program $\wp$, for each $j \in \beta_{k+1}$, where $W''_j = w'_j$ for each $j \in \gamma_{k+1}$ and $W''_j = g_{k+1}(\bar{v}_{\gamma_{k+1}})$, for each $j \in \beta_{k+1}$.

Proof: We are using the induction on k. For k = 1 the statements of the theorem result from the proposition 1. The proposition 2 specifies the case k = 2. We assume that the statement a) holds for k-1, therefore

$$g_k(\bar{v}_{\gamma_k}) \text{ is a } \varphi_{k-1}\text{-fixed point for } f\,(w'_1, ..., w'_{n+1}), \quad (4.5)$$

where $w'_j = v_j$ for every $j \in \gamma_{k-1}$ and $w'_j = g_k(\bar{v}_{\gamma_k})$ for every $j \in \beta_{k-1}$.

By the hypothesis, $g_{h+1}(\bar{v}_{\gamma_{h+1}})$ is a φ_h-fixed point of $g_h(\bar{v}_{\gamma_h})$, for every h, $0 \le h \le k$. That means, if $\bar{v}_{\gamma_h} = (i_1, ..., i_q)$, where $q = n + 1 - h$, then we have

$$g_{h+1}(\bar{v}_{\gamma_{h+1}})=g_h(t_{i_1},...,t_{i_q}) \quad (4.6)$$

where $t_{\varphi_h} = g_{h+1}(\bar{v}_{\gamma_{h+1}})$ and $t_{i_j} = v_{i_j}$, $j = \overline{1,q}$, $i_j \ne \varphi_h$.

Taking $h = k$ in the relation (4.6), it obtains

$$g_{k+1}(\overline{v}_{\gamma_{k+1}}) = g_k(t_{i_1}, \ldots, t_{i_q}); \tag{4.7}$$

where $\overline{v}_{\gamma_k} = (i_1, \ldots, i_q)$ and

$t_{\varphi_k} = g_{k+1}(\overline{v}_{\gamma_{k+1}})$, $t_{i_j} = v_{i_j}$ for every j, $1 \le j \le q$, $\varphi_k \ne i_j$, for every j, $1 \le j \le q$. (4.8)

In the relation (4.5) we consider $g_{k+1}(\overline{v}_{\gamma_{k+1}})$ instead of v_{φ_k} and using the relation (4.6), it results (4.5) for k.

The part b)of the theorem follows from a).

For two operators with the same argument which are in the $\le_t$ ordering, we are interested if for two φ_k -fixed points, one of them from the first and another from the second, the same relation between them with respect to the $\le_t$ ordering exists. The following theorem emphasizes one of these situations.

Theorem 2: Let $g_k(\overline{v}_{\gamma_k})$ and $h_k(\overline{v}_{\gamma_k})$ be two operators which are in the $\le_t$ ordering: $g_k(\overline{v}_{\gamma_k}) \le_t h_k(\overline{v}_{\gamma_k})$. Let $g_{k+1}(\overline{v}_{\gamma_{k+1}})$ be the smallest φ_k -fixed point for $g_k(\overline{v}_{\gamma_k})$ and $h_{k+1}(\overline{v}_{\gamma_{k+1}})$ be a φ_k-fixed point for $h_k(\overline{v}_{\gamma_k})$. We have: $g_{k+1}(\overline{v}_{\gamma_{k+1}}) \le_t h_{k+1}(\overline{v}_{\gamma_{k+1}})$.

For the $\le_k$ ordering a similar result follows.

Proof: Let $\beta_k = (i_1, \ldots, i_q)$ and $\varphi_k = i_p, p \in \{1, 2, \ldots q\}$.

We have the following relations:

$$g_{k+1}(\overline{v}_{\beta_{k+1}}) = g_k(v_{i_1}, \ldots, v_{i_{p-1}}, g_{k+1}(\overline{v}_{\beta_{k+1}}), v_{i_{p+1}}, \ldots, v_{i_q}) \tag{4.9}$$

$$h_{k+1}(\overline{v}_{\beta_{k+1}}) = h_k(v_{i_1}, \ldots, v_{i_{p-1}}, h_{k+1}(\overline{v}_{\beta_{k+1}}), v_{i_{p+1}}, \ldots, v_{i_q}) \tag{4.10}$$

$$g_k(v_{i_1}, \ldots, v_{i_p}, \ldots, v_{i_q}) \le_t h_k(v_{i_1}, \ldots, v_{i_p}, \ldots, v_{i_q}) \tag{4.11}$$

Taking in (4.11) $h_{k+1}(\overline{v}_{\beta_{k+1}})$ instead of v_{i_p}, we obtain

$$g_k(v_{i_1}, \ldots, v_{i_{p-1}}, h_{k+1}(\overline{v}_{\beta_{k+1}}), v_{i_{p+1}}, \ldots, v_{i_q}) \le h_{k+1}(\overline{v}_{\beta_{k+1}}) \tag{4.12}$$

Hence $g_{k+1}(\overline{v}_{\beta_{k+1}}) \le_t h_{k+1}(\overline{v}_{\beta_{k+1}})$.

Extended Stable Models for $\wp$: In the following we consider the application of fixed-point iteration on $\Psi_\wp$.

Let $f(v_1, \ldots, v_{n+1})$ be the operator $\Psi_\wp(v_1, \ldots, v_{n+1})$ defined for logic program $\wp$ with n negations. We are interested in the application of fixed point iteration on f. If we denote this iteration by *it*, then we obtain chains like this:

$$f \xrightarrow{it} g_1 \xrightarrow{it} g_2 \xrightarrow{it} \ldots \xrightarrow{it} g_m$$

In general, there exist many chains, which begin with f. The application of *it* to g_i depends on the properties of monotony or anti-monotony for g_i.

The operator g_i will be called generalized stability operator for $\wp$ providing the ordering considered is $\le_t$ for all iterations precedents.

The operator g_m depends on h arguments, where $h = n + 1 - m$. Now, we need several notations for representing the fixed points of operators.

Let g be an operator with the arguments $\overline{v} = (v_1, \ldots, v_{n+1})$ and being monotonic with respect to $\le_t$ ordering and the argument h. We denote $(t, s, h)g$ the smallest fixed point under the $\le_t$ ordering for the operator $g'(\overline{v}_h) = (\lambda x)g(v_1, \ldots, v_{h-1}, x, v_{h+1}, \ldots, v_{n+1})$.

$(t, S, h)g$ denotes the greatest fixed point under the $\le_t$ ordering for $g'(\overline{v}_h)$.

Let g be an anti-monotonic operator under the $\le_t$ ordering and the argument h. The two extreme oscillation points in this ordering for $g'(\overline{v}_h)$ are denoted $(t, \eta, h)g$ and $(t, \psi, h)g$.

For the $\le_k$ ordering the smallest fixed point and the greatest fixed point are denoted $(k, s, h)g$, $(k, S, h)g$ respectively and the two extreme oscillation points are denoted $(k, \eta, h)g$ and $(k, \psi, h)g$. The argument for all these points is $\overline{v}_h$.

Let $v_{\alpha_1}, \ldots, v_{\alpha_h}$ be the arguments of g_m. We have:

$g_m(v_{\alpha_1}, \ldots, v_{\alpha_h}) = (\sigma_m, t_m, i_m) \ldots (\sigma_1, t_1, i_1)f(v_1, \ldots, v_{n+1})$

where

$(\alpha_1, \ldots, \alpha_m) \cup (i_1, \ldots, i_m) = (1, 2, \ldots, n+1)$

$\sigma_i \in \{t, k\}, t_i \in \{s, S, \eta, \psi\}, i = 1,m.$

If the operator g_{m-1} is monotonic on $\le_t$ ordering and with respect to the argument i_m, then there exist $g'_m = (t, s, i_m)\, g_{m-1}$ and $g''_m = (t, S, i_m)\, g_{m-1}$.

In case that the operator $f = \Psi_\wp$ is monotonic on $\le_t$ ordering and with respect to the arguments $\alpha_1, \ldots, \alpha_n$, then the fixed point iteration can apply on f, by the theorem 1, which produces an operator $g_n(v_i)$, where $i \in \{1, 2, \ldots, n+1\} - \{\alpha_1, \ldots, \alpha_n\}$.

If we can obtain g_n, then each fixed-point of g_n is called an extended stable valuation for $\wp$.

Let as denote by w such a fixed-point of g_n. It results that $\overline{w} = (w, \ldots, w)$ is i-fixed point for $\wp$, for every i, $1 \le i \le n+1$. Moreover w is a model for $\wp$. This model w will be called an extended stable model for $\wp$.

Conclusion

Stable models were studied very intensively in the literature. These models come closer to capturing the meaning of logic programs with one negation that other semantics.

We have introduced extended stable model semantics of logic programs with many negations, which extends the notion of stable model semantics for logic program with one negation.

References

Da Costa, N.C.A., L.J. Henschen, J.J. Lu and V.S Subrahmanian; 1990. Automatic Theorem Proving in Paraconsistent Logics, in M.E. Stickel (ed), Proc. 10th International Conference Automated Deduction,. Springer Lecture Notes in AI, vol. 449.

Fages Francois; 1989. A New Fixpoint Semantics for General Logic Programs Compared with the Well-Founded and the Stable Model Semantics, Proc. of the Seventh International Conference of Logic Programming.

Fitting Melvin, 1993. The Family of stable models, J. Logic programming, :17:197-225.

Fitting Melvin, 1991. Well-Founded Semantics, Generalized, Proc. of the 91 International Symposium Logic Programming, 71-88.

Gelfond, M. and V. Lifschitz; 1988. The Stable Model Semantics for Logic Programming. In R.A. Kowalski and K.A. Bowen, editors, Proc. 5th International Conference and Symposium on Logic Programming, . 1070-1080, Seatle, Washington, 15-19.

Kifer, M., 1998. On the Decidability and Axiomatization of Query Finiteness in Deductive Databases. In JACM, vol. 45; no. 4, , 588- 633.

Kowalski, R. and F. Sadri; 1990. Logic Programs with Exceptions, in D. Warren and P. Szeredi (eds), Logic Programming: Proc. International Conference and Symposium.

Przymusinski Teodor, C. 1989. Extended Stable Semantics for Normal and Disjunctive Programs, Proc. of seventh Int. Conference Logic Programming.

Rajasekar, A. and J. Minker; 1989. A Stratification Semantics for General Disjunctive Programs,Proceedings of the North American Conference on Logic Programming, Overbeek(Editors), MIT Press, 573-586,.

Spyratos, N. and D. Stamate; 1996. Bases de donnees avec Informations incertaines, Semantique et mises a jour, JFPLC'96, 49-63.

Wagner Gerd, 1991. A Database Needs Two Kinds of Negation, MFDBS'91 (3rd Symposium on Mathematical Fundamentals of DB and Knowledge Base Systems), Rostock, (Springer Verlag LNCS-495).