# A FIXPOINT APPROACH TO DECLARATIVE SEMANTICS OF LOGIC PROGRAMS

**Phan Minh Dung**[1]

**Kanchana Kanchanasut**

*Division of Computer Science*
*Asian Institute of Technology*
*G.P.O. Box 2754*
*Bangkok 10501, Thailand*

## Abstract

Two new ways to specify the declarative semantics of a logic program, based on the least fixpoint of two operators associated with the program, are defined. They are a simple and natural extension of the fixpoint theory of definite Horn programs developed by van Emden and Kowalski in [5].Then we show that for locally stratified programs, these two approaches lead to the same semantics like the perfect model semantics. A new notion of sufficient stratification, which is more general than the local stratification [7], is introduced and we show that for sufficiently stratified programs, our method specifies correctly their intended semantics. While our first construction is equivalent to the stable model semantics, the second construction is properly more general than the stable model approach. We believe that the second construction is also more general than the perfect model approach. We show some evidences for our belief.

---

1. On leave from: Institute of Computer Science and Cybernetics, National Center for Scientific Research of Vietnam, Lieugiai, Badinh, Hanoi, Vietnam.

# 1. Introduction

A logic program can be viewed as a set of first order formulas and its declarative semantics can be defined by specifying one or more models of this set. For definite Horn programs, its least Herbrand model, or equivalently, the least fixed point of a certain operator associated with the program can be considered as its declarative semantics [5]. For programs with negation, these solutions do not apply directly, because the program can have many minimal Herbrand models and the corresponding operator can have many minimal fixed points.

Clark [2] has introduced the Clark's predicate completion as the declarative semantics of a general logic program. This approach, though very intuitive, has many drawbacks and does not work well in many cases [8,9,10].

In [1,4], a class of stratified programs which disallow recursion through negation, is introduced and its declarative semantics is defined by a iterated fixpoint construction.

Przymusinsky [7] has introduced the concept of perfect model semantics and proved that for a class of locally stratified programs, a unique perfect model exists which is the iterated fixpoint semantics if the program is stratified. A generalization of this aproach, the weakly perfect model semantics, is given in [8].

Another way to specify the model theoretic semantics of logic programs is the stable model approach where the declarative semantics is described by the set of stable models of the programs [3]. For locally stratified programs, this aproach leads to the same semantics as the perfect model approach.

In this paper, we propose a new way to specify the declarative semantics of a logic program by a simple and natural extension of the classic fixpoint aproach of van Emden and Kowalski [5]. The basic idea is formalized by the introduction of the notion of quasi-interpretation as a set of ground clauses of the form

$A \leftarrow \neg B_1, \ldots, \neg B_n$ where $A, B_i$ are atoms. A corresponding operator on quasi-interpretations is defined which is monotonic and continuous. The declarative semantics is then defined as the Clark's completion of the least fixed point of this operator.

It is why we call our construction the fixpoint completion of P.
The naturalness of our approach can be demonstrated by the following examples.

**Example** Let P be p(x)<— q(x)
q(a)<—

Then the declarative semantics of P is the Herbrand model {p(a),q(a)}. This model can be determined uniquely by the following formula:

$\forall x(\ p(x) \leftrightarrow x=a\ )\ \&\ \forall x(\ q(x) \leftrightarrow x=a\ )$

where & is the logical 'and'-operator.
##

**Example** Let P be p(a)<— ¬q(x)

Then the least fixpoint of our operator on quasi-interpretations is the set { p(a)<— ¬q(a) }. The declarative semantics is defined by the Clark's completion of this set which has exactly one Herbrand model {p(a)}. Clearly, {p(a)} is the intended meaning of P.
##

We show then following results:

- For locally stratified programs, our construction leads to the same semantics as the perfect model approach. Moreover, if the program is weakly stratified, then its unique weakly perfect model is also the unique Herbrand model of our construction.

- A new notion of sufficient stratification, which is more general than the local stratification, is introduced. We show that for sufficiently stratified programs, our approach specifies naturally their intended meaning.

- The equivalence between our approach and the stable model approach is proved.

- Let us consider the simple program P: p<— ¬p . It is clearly that the semantics of P is {p}. But the Clark's completion of P is inconsistent. P has also no stable model.
To handle such a program, we extend our construction in chapter 3. The new construction is called the residual fixpoint completion. Then

the relations of the new construction to the perfect model semantics is discussed. We show some evidences that the new construction may be more general than the perfect model approach.

The relations between our approaches and other approaches in the literature can be illustrated by the following diagram:

$$
\begin{tikzcd}
 & \text{model of residual fixpoint completion} & \\
\text{weakly perfect model} \arrow[ur, "?"] & \text{model of fixpoint completion} \arrow[u] \arrow[r, leftrightarrow] & \text{stable model} \arrow[ul] \\
\text{perfect model} \arrow[u] & &
\end{tikzcd}
$$

where "X-->Y" means: "X is Y".

About the relations between perfect models, weakly perfect models, stable models, see [3,7,8].

## 2.The fixpoint completion

We assume the reader's familiarity with the notions in [6]. For safety, we recall some of them in the following.

A program clause is a clause of the form

$$A \leftarrow L_1, \ldots, L_n \quad \text{with} \quad n \geq 0$$

where A is an atom and $L_i$ are literals.

A program is a set of program clauses.

From now on, P denotes an arbitrary but fixed program.The Herbrand universe (resp.the Herbrand base) of P is denoted by HUP (resp.HBP). The set of ground instances of P is denoted by Gp. We introduce now the crucial notion of quasi-interpretation.

*Definition*

A *quasi-interpretation* is a set of ground program clauses of the form $A \leftarrow \neg B1, \ldots, \neg Bn$ , $n \geq 0$, where $Bi, A$ are ground atoms.
The set of quasi-interpretations is denoted by $QI$.
##

It is clear that QI is a complete lattice wrt the set inclusion.

Let C be the ground clause
$A \leftarrow \neg B1, \ldots, \neg Bn, A1, \ldots, Am$ with $n \geq 0$, $m \geq 0$ and let Ci be ground clauses $Ai \leftarrow \neg Bi_1, \ldots, \neg Bi_{ni}$ with $1 \leq i \leq m$ and $ni \geq 0$ .

Then $T_C(C1, \ldots Cm)$ is the following clause

$$A \leftarrow \neg B1, \ldots, \neg Bn, \neg B1_1, \ldots, \neg B1_{n1}, \ldots, \neg Bm_1, \ldots, \neg Bm_{nm}$$

We introduce now the transformation Tp on quasi-interpretations

$$Tp : QI \longrightarrow QI$$

$$Tp(I) = \{ T_C(C1, \ldots, Cm) \mid C \in Gp \text{ and } Ci \in I ,\ 1 \leq i \leq m \}$$

*Theorem 1*. *Tp is continuous.*

The proof of this theorem is quite similar to the proof of the corresponding theorem in [6,page 37].
■

Let $LFP_n = Tp^n(\emptyset)$.
$LFP = \cup\{ LFP_n \mid n \geq 1 \}$ (The least fixpoint of Tp)

Let p be a predicate of P and {C1,C2,... } be the set of clauses in LFP whose heads are atoms with predicate p.

Then the *Clark's completed definition of p is*

$$\forall x( p(x) \leftrightarrow E1 \vee \ldots \vee Em \vee \ldots)$$

where the right hand side is a (possibly infinite) disjunction.

Every Ei is of the following form:

$x=t \ \& \ \neg B1 \ \& \ \ldots \& \ \neg Bn$

where Ci is a clause of the form
$p(t) \leftarrow \neg B1, \ldots, \neg Bn$ and for the sake of simplicity, we assume that the predicates of P are unary.
An infinite disjunction is true wrt an interpretation if one of its elements is true wrt this interpretation.
If there is no clauses whose heads are atoms with predicate p, then the completed definition of p is

$\forall x \ \neg p(x)$

The Clark's completion of LFP, which is called the *fixpoint completion of P, fixcomp(P)*, is the collection of the completed definitions of predicates of P together with the Clark's equality theory.

Let C be a clause. Then $C^-$ (resp. $C^+$) denotes the set of literals (resp. the atom) occurring in the body (resp. in the head) of C.

*Theorem 2* ( Basic theorem )
*a.) Every Herbrand model of P is a model of LFP.*
*b.) Every Herbrand model of the fixpoint completion of P, fixcomp(P), is a model of the Clark's completion of P, comp(P).*

**Proof**
**a.)** Let I be a Herbrand model of P. We will show by induction on n that $I \vdash C$ for any $C \in LFP_n$. It is clear that $I \vdash C$ for $C \in LFP_1$. Let $C \in LFP_n$. Then there is $D \in Gp$ and $C1, \ldots, Cm \in LFP_{n-1}$ such that $C = T_D(C1, \ldots, Cm)$. By induction hypothesis we have: $I \vdash Ci$. Let $I \vdash C^-$. We have to show that $I \vdash C^+$. From $I \vdash C^-$, it follows that for any i $I \vdash Ci^-$ holds. Therefore $I \vdash Ci^+$. That means that $I \vdash D^-$ holds. Since I is a model of D, we can conclude $I \vdash C^+$ from $I \vdash D^+$ and $D^+ = C^+$.

**b.)** Let I be a Herbrand model of fixcomp(P). First, we want to show that I is a model of P. Let $C \in Gp$ with $I \vdash C^-$. Assume that {A1,...,Am} is the set of positive literals in the body of C Because I is a model of fixcomp(P), there is a clause $Ci \in LFP$ for every Ai such that $Ci^+ = Ai$

and $I \vdash Ci^-$ . Let $D=T_C(C1,...,Cm)$. Then $I \vdash D^-$ holds. Since I is a model of LFP, we have $I \vdash C^+$ from $I \vdash D^+$ and $D^+ = C^+$ . It remains to show that for any A of I, there is a C of Gp such that $C^+ = A$ and $I \vdash C^-$ . Let $A \in I$. Then there is a D from LFP such that $A = D^+$ and $I \vdash D^-$ . Because $D=T_C(C1,...,Cm)$ for some $C \in Gp$ and $Ci \in LFP$ , we can conclude immediately that $I \vdash C^-$ and $C^+ = D^+ = A$
■

**Note** The following example shows that in general the reverse of part b, theorem 2 does not hold.

Let P be a<— a
a<— ┐b
b<— b
b<— ┐c
c<— c
c<— ┐a

The Clark's completion of P is consistent. But it does not provide us any useful information about the declarative semantics of P. Fixcomp(P) is inconsistent. This is not a weakness of fixcomp(P), but a demonstration that, in general, the Clark's completion could not specify the correct meaning of logic programs.
On other side, we believe that the way to specify the declarative meaning of a logic program by its Clark's completion is the most intuitive one. But, as shown in the literature [7,8], this method has many drawbacks. In the following examples, we demonstrate how these drawbacks can be eliminated by our fixpoint completion. The programs in these examples are taken from [10]. Throughout the paper, we assume that the Clark's equational theory axioms are satisfied.

**Example** Let P1 be the program

*natural-number(0)<—*
*natural-number(succ(x))<— natural-number(x)*

It is clear that the Clark's completion of P specifies correctly the intended meaning of P. So we could not derive the conclusion that IBM is a natural number from comp(P). However, after adding to P a neutral (meanigless) clause:

*natural-number(x)<— natural-number(x)*

The new program P2 will have a different semantics, because its Clark's completion has many models and only one of them describes the intended meaning of P2.

Let us consider the fixpoint completion of P2.

$LFP_1$ = { natural-number(0)<— }
$LFP_n$ = { natural-number($succ^i$(0)) | $0 \leq i < n$ }
LFP = { natural-number($succ^i$(0)) | $i \geq 0$ }

fixcomp(P) is the following formula:

$\forall$x( natural-number(x) <-> x=0 v x=succ(0) v...v x=$succ^i$(0) v...)

We see that fixcomp(P) describes uniquely the set of natural numbers.
##

**Example** Suppose now that P3 is given by the following clauses:

*edge(a,b)<—*
*edge(c,d)<—*
*reachable(a)<—*
*reachable(x)<— reachable(y),edge(y,x)*
*unreachable(x)<— ¬reachable(x)*

We obviously expect that vertices c,d to be unreachable and indeed, Clark's semantics implies unreachable(c),unreachable(d). However, the Clark's completion of P4 obtained from P3 by adding the clause

*edge(d,c)<—*

could not implies this conclusion, although it still appears to be expected from the given information. Now, we want to see what is provided by our fixpoint completion.

$LFP_1$ = { edge(a,b)<—
edge(c,d)<—
edge(d,c)<—
reachable(a)<— } U
{ unreachable(x)<— ¬reachable(x) | x$\in${a,b,c,d} }
$LFP_2$ = $LFP_1$ U { reachable(b) }
$LFP_3$ = $LFP_2$ = LFP

fixcomp(P) is the following formula:

$$\forall xy(\ edge(x,y) \leftrightarrow (x=a\ \&\ y=b)\ v\ (x=c\ \&\ y=d)\ v\ (x=d\ \&\ y=c)\ )$$

$$\forall x(\ reachable(x) \leftrightarrow x=a\ v\ x=b\ )$$

$$\forall x(\ unreachable(x) \leftrightarrow (\ x=a\ \&\ \neg reachable(a)\ )\ v\ (\ x=b\ \&\ \neg reachable(b)\ )\ v\ (\ x=c\ \&\ \neg reachable(c)\ )\ v\ (\ x=d\ \&\ \neg reachable(d)\ )\ )$$

It is easily to see that

$$fixcomp(P) \vdash \forall x(\ unreachable(x) \leftrightarrow x=c\ v\ x=d\ )$$

That means that the fixpoint completion implies unreachable(c) and unreachable(d). The only Herbrand model of fixcomp(P) is clearly the intended model of the program.
##

## 2.1 Relations to perfect model semantics

We recall the **priority relation** $<$ between atoms A,B in HBP and an auxiliary relation $\leq$ from [7].

1. $C<B$ , if $\neg B$ is a negative literal in the body of a clause from Gp and C is its head.
2. $C\leq A$ , if A is a positive literal in the body of a clause from Gp and C is its head.
3. If $A\leq B$ and $B\leq C$ then $A\leq C$.
4. If $A\leq B$ and $B<C$ (resp. $D<A$) then $A<C$ (resp. $D<B$).
5. If $A<B$ then $A\leq B$.
6. Nothing else satisfies $<$ or $\leq$ .

##

Let M,N be two Herbrand models of P. We say that N is **preferable** to M, **N<<M**, if for each $A\in(N-M)$ there is an $B\in(M-N)$ with $A<B$. M is **perfect** if there are no models preferable to M. More about perfect model semantics see [7].

*Basic lemma*
*Let I be a Herbrand model of fixcomp(P) and J be a Herbrand model of LFP. Then*
*$I<<J$ holds.*

**Proof**
Let $A\in(I-J)$. Then there is a C from LFP such that $A=C^+$ and $I\vdash C^-$ . From $J\vdash C$ and $A\in(I-J)$ ,

it follows that there is a negative literal $\neg B$ in the body of C with $B \notin J$ . Since I is a model of $C^-$, $B \notin (J-I)$ holds . Thus, $I<<J$ holds, because of $A<B$ .

■

Now, we are interested in programs whose fixcomp(P) has a unique Herbrand model.
A relation $<$ is called **noetherian** if there is no infinite sequence $E0<E1<E2<....$

*We say that P is <u>sufficiently stratified</u> if the priority relation of LFP is noetherian.*

<u>*Theorem 3.1*</u>
*If P is sufficiently stratified, then fixcomp(P) has a unique Herbrand model.*

**Proof** See appendix.

■

If the priority relation of P is noetherian then P is said to be locally stratified [7]. It is clear that the sufficient stratification is more general than the local stratification. A program which is sufficiently stratified but not locally stratified is given in the next chapter.

The following theorem makes explicit the relations between the fixpoint completion and the perfect model semantics.

<u>*Theorem 3.2*</u>
*Assume that fixcomp(P) is consistent and P has one Herbrand perfect model. Then fixcomp(P) has exactly one Herbrand model which is also the unique Herbrand perfect model of P.*

**Proof**
Let J be a Herbrand model of P. Then J is also a model of LFP, according to the basic theorem. Thus, $I<<_0 J$ for every Herbrand model I of fixcomp(P), following the basic lemma, where $<<_0$ is the preferable relation of LFP. Let $<_0,<_1$ be the priority relations of LFP and P, respectively. It is clear that $<_0$ is a subset of $<_1$ . Hence, for any Herbrand models K,K' of P,

$K \ll_1 K'$ if $K \ll_0 K'$. Thus, $I \ll_1 J$ holds. If J is a perfect model of P then J is clearly the unique Herbrand model of fixcomp(P).

■

*Corrollary 1*
*If P is locally stratified then fixcomp(P) has exactly one Herbrand model which is the unique Herbrand perfect model of P.*

**Proof**
In appendix, we give a proof which is more elegant and shorter than the proof in [7].

■

## 2.2. Relations to stable model semantics

Let M be a set of atoms of P. We define $S_M$ as the program obtained from Gp by deleting

i) each clause that has a negative literal $\neg B$ in its body with $B \notin M$ .
ii) All negative literals in the body of the remaining clauses.

$S_M$ is clearly a set of ground definite Horn clauses. If M coincides with the least Herbrand model of $S_M$, then M is a model of P and it is called a **stable model** of P. For more details, see [3].

*Theorem 4*
*Every Herbrand model of fixcomp(P) is a stable model of P and vice versa.*

**Proof**
"==>" Let I be a Herbrand model of fixcomp(P). It follows immediately from the basic theorem and the definition of $S_I$ that I is a model of $S_I$. It remains to show that I is the least Herbrand model of $S_I$. Let $A \in I$. Then there is a clause C from some $LFP_n$ such that $C^+ = A$ and $I \vdash C^-$ . We show by induction on n that A belongs to the least Herbrand model of $S_I$.
"n=1". That means $C \in Gp$. Since $I \vdash C^-$ holds, the unit clause A<— belongs to $S_I$.
"n-->n+1". Let D be from Gp and C1,...,Cm be from

$LFP_n$ such that $C=T_D(C1,...,Cm)$. Therefore $I \vdash Ci^-$ for any $1 \leq i \leq m$ . By induction hypothesis, the head of Ci belongs to the least Herbrand model of $S_I$. Let E be the clause obtained from D by deleting each negative literal in the body.Hence, E is a clause of $S_I$. We observe that every atom in the body of E belongs to the least Herbrand model of $S_I$. That means that the head of E which is A, belongs to the least Herbrand model of $S_I$.

"<==" Let I be a stable model of P. R is the program obtained from Gp by removing each clause that has one negative literal $\neg B$ in the body with $B \notin I$. Then it is clear that I is a model of $comp(lfp(T_R))$. Let C be a clause in $LFP-lfp(T_R)$. By induction on n with $C \notin LFP_n$, we can show that C contains one negative literal $\neg B$ with $B \notin I$ in its body. That means that I is not a model of $C^-$. Thus, I is a model of C. Since $I \vdash P$ and $I \vdash comp(lfp(T_R))$ and $lfp(T_R)$ is a subset of LFP, I is a model of comp(LFP).

■

The relation between fixpoint completion and stable model semantics is similar to the relation between circumscription and closed world assumption in the deductive database theory: while the fixpoint completion (resp.the circumscription) specifies the declarative meaning by a syntactical definition, the stable model approach (resp. the closed world assumption) describes a model or a set of models considered as the declarative semantics by a meta rule.

In [8], the class of weakly stratified programs is defined which is a extension of the notion of stratification in [1,4]. It is shown there that for this class, the stable model semantics and the weakly perfect model semantics coincide. Thus, it follows immediately the following corrollary.

*Corrollary 2.1*
*For weakly stratified programs, fixcomp(P) has exactly one Herbrand model which is the unique weakly perfect model of P*

■

From the theorem 4 and 3.2, it is immediately clear that the following corollary 2.2 holds.

*Corollary 2.2*
*If P has a stable model and a perfect model then they are the same.*
■

**Example** [3,8]
Let P be the program defined by following clauses

$p(a,b) \leftarrow$
$q(x) \leftarrow p(x,y), \neg q(y)$

P can assigned the following real life interpretation.

p(x,y) means: x,y are a couple and on every weekend trip, x is the driver of the family car.
q(x) means: x is the husband.
The second clause represents the commonsense understanding that the husband normally drives the car.

$LFP_1 = \{ p(a,b) \leftarrow \}$
$LFP_2 = \{ p(a,b) \leftarrow$
$\quad q(a) \leftarrow \neg q(b) \}$
$LFP_3 = LFP_2 = LFP$

fixcomp(P): $\forall x,y\ ( p(x,y) \leftrightarrow x=a \ \&\ y=b )$
$\forall x\ (q(x) \leftrightarrow x=a \ \&\ \neg q(b) )$

fixcomp(P) has only one Herbrand model {p(a,b),q(a)} which is also clearly the unique stable model of P.
##

# 3. The residual fixpoint completion

There are programs that have a well intended semantics, but their fixpoint completions are inconsistent. In the following, we give some examples of these programs and discuss the way to specify the declarative meaning of them by an extension of our fixpoint completion.

**Example 1.** Let P be $\quad a \leftarrow b$
$\quad b \leftarrow \neg a$

{a} is clearly the intended semantics of P.

The least fixpoint of $T_P$ is $\{\ a\leftarrow \neg a$
$b\leftarrow \neg a\ \}$

fixcomp(P) is inconsistent. But we observe that by transformation the clause $a\leftarrow \neg a$ into the unit clause $a\leftarrow$ , we get a new program whose completion is consistent and specifies the intended meaning.
##

**Example 2.** Let P be $a\leftarrow b$
$b\leftarrow \neg b$

This program has only one Herbrand model {a,b} which is of course the intended semantics of P.

$LFP_1 = \{\ b\leftarrow \neg b\ \}$
$LFP_2 = \{\ b\leftarrow \neg b$
$a\leftarrow \neg b\ \}$
$LFP = LFP_2$

fixcomp(P) is inconsistent. Moreover, the Clark's completion of the program obtained by transforming the clause $b\leftarrow \neg b$ into the unit clause $b\leftarrow$ , does not specify the intended semantics. But we can observe following. First, we transform $LFP_1$ into $\{b\leftarrow\ \}$ and then applying $T_P$ on $\{b\leftarrow\ \}$, we would get $\{\ b\leftarrow$
$a\leftarrow\ \}$
whose completion is clearly the intended semantics.
##

In the following, we define a modification of the fixpoint completion based on the idea discussed in the examples. Then we discuss the relations of the new construction to the fixpoint completion and perfect model semantics.

Let C be a ground clause with $C^+ = A$. The clause obtained from C by deleting each negative literal $\neg A$ in the body of C, is called *the residuent of C, resC . The residuent of a set S of ground clauses, resS*, is the set of residuents of clauses of S.

A new transformation $R_P$ on quasi-interpretations is defined as follows:

$$Rp : QI \dashrightarrow QI$$

$$Rp(I) = res(Tp(I))$$

*Theorem 5* *Rp is continuous.*

**Proof**
It is easy to see that res:QI-->QI is continuous. Theorem 5 follows directly from this fact and theorem 1 .
■

For the sake of simplicity, we introduce the following notions:

LFR = lfp(Rp) ( the least fixpoint of Rp )
$LFR_n = Rp^n(\phi)$

*The residual fixpoint completion of P is defined by:*

*rfixcomp(P) = comp(LFR)*

We will show that the declarative semantics of logic programs can be specified by rfixcomp(P).

**Example 3**
Let P be the program in example 1 . Then we have
$LFR_1$ = { b<— ┐a }
$LFR_2$ = { b<— ┐a
a<— }
LFR = $LFR_2$
The only Herbrand model of rfixcomp(P) is {a} (the intended model).
##

**Example 4**
Let P be the program in example 2. Then we have
$LFR_1$ = { b<— }
$LFR_2$ = { b<—
a<— }
rfixcomp(P) has only one Herbrand model {a,b} (the intended semantics of P).
##

The following theorem justifies the correctness of rfixcomp(P).

*Theorem 6 ( Second basic theorem )*
*1. Every Herbrand model of P is a model of LFR.*
*2. Every Herbrand model of rfixcomp(P) is a model of P.*

**Proof**

1. Let I be a Herbrand model of P, $C \in LFR$ and $I \vdash C^-$. We have to show: $A \in I$ where $C^+ = A$. The proof is by induction on n with $C \in LFR_n$.

"n=1" There is $D \in Gp$ such that $D^- - \{\neg A\} = C^-$ where $A = D^+$. Assume that not($A \in I$) holds. Then $I \vdash D^-$. Since I is a model of P, we have $A \in I$. Contradiction. That means: $A \in I$.

"n==>n+1" Let $C = resT_D(C1, \ldots, Cm)$ with $D \in Gp$ and $Ci \in LFR_n$. Assume that not($A \in I$) holds. Then $\forall i\ I \vdash Ci^-$. By induction hypothesis, we have: $I \vdash D^-$. Since I is a model of P, $D^+ \in I$ holds. But, $D^+ = A$. Contradiction. That means: $A \in I$.

2. Let I be a Herbrand model of rfixcomp(P) and $C \in Gp$ such that $I \vdash C^-$. Let C be $A \leftarrow \neg B1, \ldots, \neg Bn, A1, \ldots, Am$. From $I \vdash C^-$, it follows that $Ai \in I$ for any $1 \leq i \leq m$. Since I is a model of comp(LFR), there is for each Ai a clause Ci from LFR such that $Ai = Ci^+$ and $I \vdash Ci^-$. Let $D = resT_C(C1, \ldots, Cm)$. Then clearly $D \in LFR$ and $I \vdash D^-$ hold. Thus, $A \in I$ follows immediately from $I \vdash D$ and $D^+ = A$. That means I is a model of P.

■

The following theorem shows the relation between rfixcomp(P) and fixcomp(P).

*Theorem 7*

*If I is a Herbrand model of fixcomp(P) then I is also a model of rfixcomp(P).*

**Proof**

We define inductively a sequence of relations $(\mu_n)_n$ as follows:

i) $\mu_1 = \{ \langle C, resC \rangle \mid C \in LFP_1 \}$

ii) $\mu_{n+1} = \mu_n \cup \{ \langle C, C' \rangle \mid$ There are $D \in Gp$ and $C1, \ldots, Cm \in LFP_n$ such that $C = T_D(C1, \ldots, Cm)$ and $C' = resT_D(C1', \ldots, Cm')$ with $\langle Ci, Ci' \rangle \in \mu_n \}$

Let $\mu = \cup \{\mu_n \mid n \geq 1\}$

By induction on n, we can prove easily the following property

$\forall \langle C, D \rangle \in \mu : ( I \vdash C^- \Longrightarrow I \vdash D^- )$ and $( C^+ = D^+ )$.

The result follows immidiately from this property.

■

Analogous results like the basic lemma and the theorems 3.1,3.2 in previous chapter illuminate the relation between residual fixpoint completion and the perfect model semantics. Moreover, we are interested on the question:

*Is rfixcomp(P) consistent if P has a unique perfect model ?*

A subproblem of this problem is the following question:

*Is rfixcomp(P) consistent if P has a least Herbrand model[1] ?*

We could not give a complete answer to this question at the moment. But we believe that these quetions could be answered positively. The following results justify our belief.

*Theorem 8*

*Let L be the least Herbrand model of P. Suppose that P satisfies the following condition*

*(*) For each ground atom A, if $A \in L$, then A does not occur positively in the body of any clause of Gp.*

*Then L is the only Herbrand model of rfixcomp(P).*

**Proof**

Let Q be the program obtained from Gp by deleting all negative literals $\neg B$ with not($B \in L$) from the bodies of clauses of Gp. It is clear that every model of Q is also a model of Gp. Therefore, L is also the least Herbrand model of Q. The theorem is a direct consequence of the following lemma.

Lemma **Let $A \in L$ such that the unit clause A<- does not belong to Q. Then there is a AND-tree[2] of A wrt Q such that all its leaf nodes are $\neg A$ .**

**Proof** Let L'= L-{A}. Then L' is not a model of Q. Therefore, there is a clause $C \in Q$ with $L' \not\models C$ which is equivalent to:

$L' \models C^-$ and not($C^+ \in L'$) .

Assume that $L \models C^-$ holds. From the definition of Q and condition (*), we can conclude that the body of C is empty and A is the head of C. That means that the unit clause A<- belongs to Q.

Contradiction! Thus, L is not a model of $C^-$. It follows directly from the definition of Q that $C^-$ contains only positive literals B with not($B \in L$) and negative literals $\neg B$ with $B \in L$ . Since $L' \vdash C^-$ holds, we have $C^- = \neg A$ . The head of C can only either A or some atom B with not($B \in L$), because of not($C^+ \in L'$).

1.Case: $C^+ = A$ . That means C $A \leftarrow \neg A$ . The lemma holds.

2.Case: $C^+ = B$ with not($B \in L$) . Let $L_1 = L' \cup \{B\}$ . We construct inductively a sequence $(L_i)_i$ as following:

$$L_{i+1} = \begin{cases} \phi, & \text{if there is } D \in Q \text{ such that } L_i \vdash D^- \text{ and } D^+ = A \\ L_i \cup \{ B \mid \text{there is } D \in Q : B = D^+ \text{ and } L_i \vdash D^- \} & \text{otherwise.} \end{cases}$$

The lemma holds clearly if there is a empty Li. Assume that $L_i$ is not empty for every i. Let LH = $\cup\{L_i \mid i \geq 1\}$. We will show that LH is a model of Q. Let $D \in Q$ with $LH \vdash D^-$ . From the definition of Q and LH, we have:

a) the only negative literal in $D^-$ is $\neg A$ and

b) if B is a positive literal in $D^-$ then $B \in (LH-L)$.

$D^-$ is finite. Thus, there is an i such that each positive literal in $D^-$ belongs to $L_i$ . Consequently, $L_i \vdash D^-$ holds. Because $L_{i+1}$ is not empty, $D^+$ and A are different. If $D^+ \in L'$ then $D^+ \in LH$. If not($D^+ \in L'$) then $D^+ \in L_{i+1}$. That means $LH \vdash D$ . Since L is the least Herbrand model of Q, L is a subset of LH. Contradiction, because $A \in L$ and not($A \in LH$)!

■

Assume that we could answer the above questions positively. Then the following condition is necessary.

*($) $\forall A \in L \quad \exists C \in LFR : \quad C^+ = A$*

*where L is the least Herbrand model of P.*

*In appendix, we give the proof of this property.*

## Appendix

A program P is called locally stratified if it is possible to decompose the Herbrand base HBP of P into disjoint sets, called strata $H_0, H_1, \ldots, H_\mu, \ldots$ where $\mu < ß$ and ß is a countable ordinal, so that for every clause

A<— ¬B1,...,¬Bn,A1,...,Am

in Gp we have that:

a.) all positive premises Ai belong to
U{ Hj | j≤k }

b.) all negative premises Bi belong to
U{ Hj | j<k }

where Hk is the stratum which contains A.

In [7], the following property is proved:

*(@) P is locally stratified iff its priority relation < is noetherian.*

More about locally stratified programs, see [7].

**Proof of theorem 3.1**

We prove first that if fixcomp(P) is consistent then it has only one Herbrand model. Assume that fixcomp(P) has two different Herbrand models I,I'. Let $A_0 \in (I-I')$. Then, according to the basic lemma in chapter 2, there is a $A_1 \in (I'-I)$ such that $A_0 < A_1$. Continuing this process infinitely, we would get an infinite increasing chain $(A_i)_i$ : $A_i < A_{i+1}$ . Contradiction to the noertherian property of <.

It remains to show that fixcomp(P) has a Herbrand model. According to (@), there is a partitioning of HBP into strata H0,H1,...,Hμ,... μ<ß . Let $H^\mu$= U{ Hi | i≤μ } and Fμ be the subset of LFP whose atoms are in $H^\mu$. It follows directly that LFP = U{ Fμ | μ<ß } and Fμ is a subset of F£ if μ≤£. Let L0={ A∈H0 | A<-∈ F0 } and Lμ = Nμ U { A∈Hμ | there is a C∈Fμ : $C^+$=A and Nμ⊢$C^-$ } where Nμ = U{ L£ | £<μ }.

We want to prove inductively on μ that Lμ is a Herbrand model of comp(Fμ).

Let μ=0. It is clear that L0 is a model of comp(F0).

Assume that L£ is a Herbrand model of comp(F£) for all £<μ. It is clear that Lμ is a model of Fμ. It remains to show that for every atom A∈Lμ, there is a clause C∈Fμ with A as its head and Lμ⊢$C^-$ . Let A∈Lμ.

1.Case: A∈L£ for some £<μ. Then there is C∈F£ so that $C^+$=A and L£⊢$C^-$ . Let At be the set of atoms occurring in $C^-$. Then At is a subset of

$H^{£}$-L£. Since $(L\mu - L£) \cap H^{£} = \phi^{3}$, we obtain:
At $\cap$ $L\mu = \phi$. That means $L\mu \vdash C^-$ .
2.Case: $A \in (L\mu - N\mu)$ . Then, according to the definition of $L\mu$, there is a clause $C \in F\mu$ : $C^+ = A$ and $N\mu \vdash C^-$ . Similarly to the first case, we can show that: $L\mu \vdash C^-$ .
From fixcomp(P) = comp(Fß), it follows immediately that L = Lß is a Herbrand model of fixcomp(P).

■

**Proof of corrollary 1**
Let <,<' be the priority relations of P,LFP, respectively. Following property (@), < is northerian. Since <' is a subset of < , <' is also noetherian. Because of theorem 3.1 fixcomp(P) has exactly one Herbrand model. Since < is noetherian, there are no different Herbrand models I,J of P such that I<<J and J<<I. Therefore, from the basic theorem and basic lemma in chapter 2, the unique Herbrand model of fixcomp(P) is also the unique perfect Herbrand model of P.

■

*<u>Property</u>   Let L be the least Herbrand model of P. Then,*
*($)   $\forall A \in L \;\; \exists C \in LFR: \;\; A = C^+$*

**Proof.**   Let At = { A | $\forall C \in LFR$: $(C^+ \neq A)$ }. Let S be the program obtained from Gp by deleting
i) each clause which has a positive literal A with $A \in At$ in the body,
ii) all negative literals $\neg A$ with $A \in At$ from the remaining clauses.
Let HS be the Herbrand base of S. Let C be a clause in S such that $C^+ \in At$. From the definition of S, it follows immediately that for every positive literal A in the body of C there is a clause D in LFR with $D^+ = A$. Since LFR is a fixpoint of Tp, there exists a clause in LFR whose head is $C^+$. Contradiction! That means that HS $\cap$ At = $\phi^{3}$. It is easily to see that every Herbrand model of S is also a model of Gp. Let I be a Herbrand model of S. Since L is the least Herbrand model of Gp and I is a model of Gp, L is a subset of I. Then it follows immediately that L is a subset of HS. We obtain now: L $\cap$ At = $\phi^{3}$.

■

# Conclusion

We have described two natural ways to specify the declarative semantics of logic programs and discussed their relations to other approaches in the literature.
But we haved considered only Herbrand-semantics. As shown in [10], Herbrand-semantics is not correct wrt universal queries. [10] also shows that the class of (not necessary Herbrand) perfect models specifies correctly the declarative semantics of logic programs and there is no problems with universal queries wrt to this semantics. An extension of our fixpoint completion is given in [11] and it is proved there that for stratified programs our natural semantics is equivalent to the perfect model semantics in [10].

# Acknowlegdements

We thank Huynh Ngoc Phien and Tomonori Kimura for their assistance and support during the time in which this paper is written. We would like also to thank many referees for their useful comments and amendments.

# Endnotes

1. From the theorem 3.2 in [8], it is likely that these two questions are equivalent.
2. An AND-tree of A wrt Q is a tree satisfying following conditions:
a) The non-leaf nodes are ground positive literals.
b) The root is A.
c) Let B be a non-leaf node. Then there is a clause C with head B from Q such that the children of B are literals in $C^-$.
d) Let L be a leaf node. Then either L is negative or L is positive such that there are no clauses in Q with head L.
3. "X n Y" denotes the intersection of the sets X and Y.

# References


[1]. K.R.Apt,H.Blair,A.Walker " *Towards a theory of declarative knowledge", in Foundation of deductive databases and logic Programming, ed. J.Minker, Los Altos,CA,1988,89-148.*

[2]. K.L.Clark. *"Negation as failure", in Logic and database, ed. H.Gallaire and J.Minker, Plenum, New York, 299-322.*

[3]. M.Gelfond,V.Lifschitz. *"The stable model semantics for logic programs", in "Fifth conference on logic programming", ed. R. Kowalski and K.A. Bowen, 1988.*

[4]. A.V.Gelder *"Negation as failure using tight derivations for general logic programs", in J. of logic programming,Vol 6,No 1/2,1989.*

[5]. M.H.van Emden,R.A.Kowalski *"The semantics of predicate logic as a programming language", JACM,Vol 23(4),1976.*

[6]. J.W.LIoyd,*"Foundation of logic programming", 2rd edition, Springer Verlag,1987.*

[7]. T.C.Przymusinski *"On the declarative semantics of deductive database and logic programming", in Foundation of deductive database and logic programming, ed. J. Minker,Morgan Kaufmann 1988.*

[8]. H.Przymusinska,T.C.Przymusinski *"Weakly perfect model semantics for logic programs", in Fifth international conference on logic programming", ed. R.A.Kowalski, K.A.Bowen 1988.*

[9]. T.C. Przymusinski " *On the declarative semantics and procedural semantics of logic programs", to appear.*

[10] T.C. Przymusinski " *Perfect model semantics" in Fifth international conference on logic programming*

[11] Phan Minh Dung " *On natural semantics of general logic programs" To appear.*