人　工　知　能　83－2
（1992．　7．　10）

Least Fixpoint Semantics of Generalized Logic Programs

Kiyoshi Akama

Faculty of Engineering, Hokkaido University

GLP theory is an axiomatic theory of logic programs, which are sets of clauses consisting of abstract atoms which are defined by an abstract axiom. Such programs are called **generalized logic programs**. In this paper we give the **least fixpoint semantics** of generalized logic programs, which has the following characteristics. First, this theory is **very general** and is **widely applicable**. Second, we use **lower continuous** functions rather than **continuous** functions used in the usual theory. Lower continuity, which is more general than continuity, is enough to develop the least fixpoint theory here. Third, in the usual theory one-step-inference transformation T_P is used to define fixpoints. But the fixpoints of T_P are **meaningless**. Instead we use knowledge increasing transformation $K_P = T_P + I_d$, which results in a general and elegant theory of the least fixpoint semantics.

一般化論理プログラムの最小不動点意味論

赤間 清

北海道大学　工学部　情報工学科

原子論理式の具体的な形を決めずに、**抽象的な公理系**によって必要な最小限の性質だけを要請して、(抽象的な) アトムなどを定義し、それをもとにして一般化された論理プログラムを導入する。このようにして公理的な論理プログラムの理論（GLP の理論）を作ることができる。それは、Prolog や制約論理型言語や typed Prolog などの論理型言語だけでなく、関数型言語やオブジェクト指向言語、さらには、ユニフィケーション文法などの宣言的側面を基礎づける理論に発展しつつある。本論文では、一般化論理プログラムの最小不動点意味論を与える。その特徴の第一は、非常に一般的な理論であり、広範囲に適用できることである。第二に、連続性ではなく、下連続性を基礎にしている。第三に、推論結果を与える変換 T_P ではなく、推論による知識の増加を与える変換 K_P を基礎にして、その不動点などを論じている。それは、従来の理論で用いられているにもかかわらず、T_Pの不動点には意味がないからである。K_P を用いることにより、一般性があり、エレガントな理論が構築できるのである。

1 Introduction

Instead of starting with usual concrete definitions of atoms and substitutions, we adopt abstract definitions of them and are constructing an **axiomatic theory** of logic programming. We call it **GLP theory** (theory of **generalized logic programs**).

The basic axiom is called **specialization systems**, which is defined as follows:

Definition 1 A *specialization system* is a 4-tuple $<\mathcal{A}, \mathcal{G}, \mathcal{S}, \mu>$ that satisfies the following conditions.

1. $\mu : \mathcal{S} \to partial\_map(\mathcal{A})$
2. $\forall s_1, s_2 \in \mathcal{S}, \exists s \in \mathcal{S}:$ $\mu(s) = \mu(s_2) \circ \mu(s_1)$
3. $\exists s \in \mathcal{S}, \forall a \in \mathcal{A} : \mu(s)(a) = a$
4. $\mathcal{G} \subset \mathcal{A}$

Elements in $\mathcal{A}$ are called *atoms*. $\mathcal{G}$ is called the *interpretation domain*. Elements in $\mathcal{S}$ are called *specializations*. The specializations that satisfy the third condition are called *identity specializations*.

Several examples of specialization systems are found in [3]. Here we show one of such examples : $\Gamma_5 =< \mathcal{A}_5, \mathcal{G}_5, \mathcal{S}_5, \mu_5 >$, where

1. $\mathcal{A}_5 = \{x, y, z, p, q, r, s, t\}$
2. $\mathcal{G}_5 = \{p, q, r, s, t\}$
3. $\mathcal{S}_5 = \{e, a, b, c, o\}$
4. μ_5 is a mapping from $\mathcal{S}_5$ to *partial_map* ($\mathcal{A}_5$) which is defined by :

$$\mu_5(e) =< \mathcal{A}_5, \mathcal{A}_5, \{(a, a) \mid a \in \mathcal{A}\} >$$
$$\mu_5(a) =< \mathcal{A}_5, \mathcal{A}_5, \{(x,p), (y,q), (z,r)\} >$$
$$\mu_5(b) =< \mathcal{A}_5, \mathcal{A}_5, \{(z,p)\} >$$
$$\mu_5(c) =< \mathcal{A}_5, \mathcal{A}_5, \{(x,q), (y,s)\} >$$
$$\mu_5(o) =< \mathcal{A}_5, \mathcal{A}_5, \{\, \} >$$

We have defined generalized logic programs on specialization systems [3].

Definition 2 A *logic program on a specialization system* Γ is a (possibly infinite) set of program clauses on $\mathcal{A}$. A program clause is of the form : $H \leftarrow A_1, \ldots, A_n$, where $H, A_1, \ldots, A_n$ are atoms in $\mathcal{A}$.

GLP theory does not refer to predicates, variables, constants, functions and substitutions which are the basic components of ordinary logic programs.

In [4], we have discussed the minimal model semantics of generalized logic programs. In this paper we will give the theory of least fixpoint semantics of generalized logic programs. See the figure in appendix. The figure shows our plan to develop GLP theory.

In the usual theory of logic programs, for example [10], the fixpoint theory is discussed based on the one-step-inference transformation T_P. However the knowledge increasing transformation $K_P = T_P + I_d$ plays the central role here. We should use the fixpoints of K_P rather than the ones of T_P. X is a fixpoint of K_P iff we can not increase X by the inference of P. But fixpoints of T_P do not have such good interpretations.

2 Fixpoint Theory

2.1 Introduction

In this section we give a fixpoint theory on a complete lattice. The theory is independent of the axioms of specialization systems, and has the following characteristics. The theory is different from the usual theories, such as [10], of logic programs, in the following points.

- We use *lower continuous* functions rather than *continuous* functions used in the usual theory. Lower continuity, which is more general than continuity, is enough to develop the least fixpoint theory here. GLP theory tries to find the most general preconditions to discuss the main structure of generalized logic programs.

- We use an increasing, monotonic and lower continuous transformation $K = T + I_d$ instead of a monotonic and lower continuous transformation T in the usual theory, because the least fixpoint of T is meaningless when it is applied to logic programs in section 3.

2.2 Basic Definitions

We begin with the definition of a complete lattice.

Definition 3 A relation $\leq$ on S is a *partial order* iff it satisfies the following conditions:

1. **[reflexive]**
   $x \leq x$, for all x in S.
2. **[asymmetric]**
   $x \leq y$ and $y \leq x$ implies $x = y$, for all x and y in S.
3. **[transitive]**
   $x \leq y$ and $y \leq z$ implies $x \leq z$, for all x,y and z in S.

We often denote $x \leq y$ by $y \geq x$.

Definition 4 S is a *partially ordered set* iff S is associated with a partial order on S.

For every subset X of a partially ordered set S, *least upper bound* of X, which is denoted by $lub(X)$, is unique if it exists. Similarly, *greatest lower bound* of X, which is denoted by $glb(X)$, is unique if it exists.

Definition 5 A partially ordered set L is a *complete lattice* iff there are $lub(X)$ and $glb(X)$ for any subset X of L.

From now on we assume that L is a complete lattice, $\leq$ is a partial order on L, $F : L \rightarrow L$ is a mapping on L, $\bot$ is the minimal element in L, and N is the set of all non-negative integers.

Definition 6 F is *increasing* iff $x \leq F(x)$ for any element x in L.

Definition 7 F is *monotonic* iff $x \leq y$ implies $F(x) \leq F(y)$ for any element x, y in L.

Definition 8 A sequence $\{x_n\}(n \in N)$ is on L iff $x_n \in L$ for all $n \in N$. A sequence $\{x_n\}(n \in N)$ on L is *increasing* iff $x_n \leq x_{n+1}$ for all $n \in N$.

Definition 9 F is *lower continuous* iff $F(lub\{x_n\}) = lub\{F(x_n)\}$ for any increasing sequence $\{x_n\}(n \in N)$ on L.

Note that continuous mappings are monotonic [10], but lower continuous mappings are not.

2.3 Fixpoint Theory on Complete Lattice

In this section we discuss the foundation of fixpoint theory of a mapping F on a complete lattice L. In most definitions and propositions here, we assume, implicitly or explicitly, that F is increasing. Increasingness is the most important assumption here.

Definition 10 $a \in L$ is a fixpoint of F iff $F(a) = a$.

Definition 11 For any $F : L \rightarrow L$, $F^\omega : L \rightarrow L$ is defined by

$$\forall x \in L : F^\omega(x) = lub\{F^n(x) \mid n \geq 0\},$$

where $F^0(x) = x$ and $F^1(x) = F(x)$.

Proposition 1 If $F(x) = x$ then $F^\omega(x) = F(x) = x$.
Proof From $F(x) = x$, $F^n(x) = x$ for all $n \in N$. Then,

$$\begin{aligned} F^\omega(x) &= lub\{F^n(x) \mid n \geq 0\} \\ &= lub\{x \mid n \geq 0\} \\ &= x \end{aligned}$$

Proposition 2 K^ω is increasing, that is, $K^\omega(x) \geq x$ for all x in L.
Proof Using $F^0(x) = x$,

$$\begin{aligned} F^\omega(x) &= lub\{F^n(x) \mid n \geq 0\} \\ F^\omega(x) &\geq F^n(x) \qquad \forall n \geq 0 \\ F^\omega(x) &\geq x \end{aligned}$$

Proposition 3 If F is monotonic, then F^ω is also monotonic.
Proof Assume $y \geq x$. As F is monotonic, $\forall n \geq 0 : F^n(y) \geq F^n(x)$. Using this,

$$\begin{aligned} F^\omega(y) &= lub\{F^n(y) \mid n \geq 0\} \\ F^\omega(y) &\geq F^n(y) \qquad \forall n \geq 0 \\ F^\omega(y) &\geq F^n(x) \qquad \forall n \geq 0 \end{aligned}$$

This shows that $F^\omega(y)$ is a upper bound of $\{F^n(x) \mid n \geq 0\}$, then

$$F^\omega(y) \geq lub\{F^n(x) \mid n \geq 0\} = F^\omega(x).$$

Proposition 4 If F is increasing and lower continuous, then $F^\omega(x)$ is a fixpoint of F for any element x in L.
Proof As F is increasing, $\{F^n(x)\}(n \geq 0)$ is also increasing. Since F is lower continuous,

$$F(lub\{F^n(x) \mid n \geq 0\}) = lub\{F(F^n(x)) \mid n \geq 0\}$$

Using these we obtain the following:

$$\begin{aligned} F(F^\omega(x)) &= F(lub\{F^n(x) \mid n \geq 0\}) \\ &= lub\{F(F^n(x)) \mid n \geq 0\} \\ &= lub\{F^n(x) \mid n \geq 1\} \\ &= lub\{F^n(x) \mid n \geq 0\} \\ &= F^\omega(x) \end{aligned}$$

This means that $F^\omega(x)$ is a fixpoint of F.

2.4 Fixpoint Theory for K

We assume that $K : L \rightarrow L$ is **increasing, monotonic and lower continuous.** The following propositions are straightforward from propositions 2, 3 and 4.

Proposition 5 K^ω is monotonic.

Proposition 6 $K^\omega(x)$ is a fixpoint of K for any element x in L.

Let x be an element in L. We denote by $Fp(K, x)$ the set of all elements in L which are fixpoints of K and more than or equal to x, that is, $y \in Fp(K, x)$ iff $K(y) = y$ and $y \geq x$. We denote the minimal element in $Fp(K, x)$ by $lfp(K, x)$.

The following is one of the most important theorems in this paper.

Theorem 1 $lfp(K, x) = K^\omega(x)$ for any x in L.
Proof From proposition 6, $K^\omega(x)$ is a fixpoint of K. From proposition 2, $K^\omega(x) \geq x$. Next, assume that y is any element in $Fp(K, x)$. From the definition, $K(y) = y$ and $y \geq x$. From proposition 5, K^ω is monotonic. Hence from $x \leq y$, $K^\omega(x) \leq K^\omega(y)$. From Proposition 1, $K^\omega(y) = y$. Therefore, $K^\omega(x) \leq y$. Then $K^\omega(x)$ is proved to be $lfp(K, x)$.

This theorem states that the least elements in the set of all fixpoints of K which are greater than or equal to x is given by $K^\omega(x)$. The usual fixpoint of K is a special case of this, because it is identical with the least elements of the set of all fixpoints of K which are greater than or equal to $\perp$.

2.5 Making K from T

We have discussed the fixpoint theory based on a increasing, monotonic and lower continuous mapping K. In this section we give one way to make a increasing, monotonic and lower continuous mapping K from a

monotonic and lower continuous mapping T. Some of the proofs are given in the appendix.

Definition 12 For every element x,y in L, we define $x + y = lub\{x, y\}$.

Definition 13 Let T_1 and T_2 be mappings on L. The mapping $T_1 + T_2$ on L is defined by

$$\forall x \in L : [T_1 + T_2](x) = T_1(x) + T_2(x).$$

Lemma 1 If L's elements x, y, z and w satisfy $x \leq y$ and $z \leq w$, then $x+z \leq y+w$.

Lemma 2 For any sequence $\{x_n\}$,$\{y_n\}$ on L, $lub\{x_n\} + lub\{y_n\} = lub\{x_n + y_n\}$.

Proposition 7 If T_1 and T_2 are monotonic, then $T_1 + T_2$ is also monotonic.

Proposition 8 If T_1 and T_2 are lower continuous, then $T_1 + T_2$ is also lower continuous.

Lemma 3 Let I_d be the identity mapping on L. Then, I_d is monotonic and lower continuous.

Theorem 2 Let $T : L \to L$ be a monotonic and lower continuous mapping. Let $K : L \to L$ be defined by $K = T + I_d$. Then, K is increasing, monotonic and lower continuous.
Proof From the assumptions and lemma 3, T and I_d are both monotonic and lower continuous. From proposition 7 and 8, K is also monotonic and lower continuous. Since $K(x) = T(x) + I_d(x) = T(x) + x \geq x$, K is increasing.

2.6 Least Fixpoint Theory

Definition 14 $a \in L$ is a least fixpoint of F iff a is a fixpoint of F and $a \leq b$ for any fixpoint b of F.

The least fixpoint of F is uniquely determined if it exists. We denote the least fixpoint of F by $lfp(F)$ if it exists. The following proposition is obvious.

Proposition 9 When there is the least fixpoint of F, $lfp(F) = lfp(F, \perp)$

Definition 15 $F \uparrow \omega$ is defined as $F^{\omega}(\perp)$.

From these we can get a special case of theorem 1.

Theorem 3 If $K : L \to L$ is increasing, monotonic and lower continuous, then

$$lfp(K) = K \uparrow \omega$$

Proof Let x be $\perp$ in theorem 1. Then, $lfp(K, \perp) = K^{\omega}(\perp)$. Therefore, $lfp(K) = K \uparrow \omega$ is obvious from proposition 9 and definition 15.

2.7 Least Fixpoint for T

The usual theory of logic programming uses the fixpoint of T, instead of K. So we give also the theory of the least fixpoint of T. However this does not mean that the least fixpoint theory of T is absolutely necessary for GLP theory. In fact, we can omit the part of discussion of the least fixpoint of T from this paper to give the fixpoint semantics of generalized logic programs. The main reason why we give the theory for T in this and the next sections is to make clear the relation between GLP theory and usual theory of logic programming.

In this and the next section we assume that T is **monotonic and lower continuous**.

Proposition 10 $\{T^n(\perp)\}(n \geq 0)$ is increasing.
Proof We prove $T^{n+1}(\perp) \geq T^n(\perp)$ for any $n \geq 0$ by induction. When n = 0, $T(\perp) \geq \perp$ is obvious since $\perp$ is the minimal element in L. Assume that the result holds

when $n = k$, that is, $T^{k+1}(\bot) \geq T^k(\bot)$. Applying the monotonic transformation T to both sides of the inequation, we get
$T^{k+2}(\bot) \geq T^{k+1}(\bot)$.
This shows that the result also holds when $n = k + 1$.

Proposition 11 $T^\omega(\bot)$ is a fixpoint of T.
Proof From proposition 10, $\{T^n(\bot)\}(n \geq 0)$ is increasing. Since T is lower continuous,

$$T(lub\{T^n(\bot) \mid n \geq 0\}) = lub\{T(T^n(\bot)) \mid n \geq 0\}$$

Using this we obtain the following:

$$\begin{aligned} T(T^\omega(\bot)) &= T(lub\{T^n(\bot) \mid n \geq 0\}) \\ &= lub\{T(T^n(\bot)) \mid n \geq 0\} \\ &= lub\{T^n(\bot) \mid n \geq 1\ \} \\ &= lub\{T^n(\bot) \mid n \geq 0\} \\ &= T^\omega(\bot). \end{aligned}$$

This means that $T^\omega(\bot)$ is a fixpoint of T.

Proposition 12 $T^\omega(\bot)$ is a least fixpoint of T.
Proof Assume that y is a fixpoint of T. We prove that $y \geq T^n(\bot)$ for all $n \geq 0$ by induction. When $n = 0$, $y \geq T^0(\bot)$ is obvious since it is equivalent to $y \geq \bot$. Next we assume that $y \geq T^k(\bot)$. By applying monotonic transformation T to both sides of the inequation, we get $T(y) \geq T^{k+1}(\bot)$. Since y is a fixpoint of T, we know $y \geq T^{k+1}(\bot)$. Thus, $y \geq T^\omega(\bot)$ for any fixpoint y of T. From proposition 11, we can conclude that $T^\omega(\bot)$ is the least fixpoint of T.

From this proposition we have T version of the theorem 3.

Theorem 4 If $T : L \rightarrow L$ is monotonic and lower continuous, then

$$lfp(T) = T \uparrow \omega$$

Proof Obvious.

2.8 $lfp(K)$ and $lfp(T)$

We will examine here the relation between the fixpoint of K and the fixpoint of T. Please remind that:

- T is monotonic and lower continuous.
- $K = T + I_d$
- K is increasing, monotonic and lower continuous.

First we prove that $K^n(\bot) = T^n(\bot)$.

Proposition 13 For all n in N,
$K^n(\bot) = T^n(\bot)$.
Proof We prove that $K^n(\bot) = T^n(\bot)$ for all $n \geq 0$ by induction. When $n = 0$, $K^0(\bot) = T^0(\bot)$ is obvious since it is equivalent to $\bot = \bot$. Next we assume that $K^k(\bot) = T^k(\bot)$, and prove $K^{k+1}(\bot) = T^{k+1}(\bot)$.

$$\begin{aligned} &K^{k+1}(\bot) \\ &= K(K^k(\bot)) \\ &= K(T^k(\bot)) \quad \text{by induction} \\ &= T(T^k(\bot)) + T^k(\bot) \\ &= T^{k+1}(\bot) + T^k(\bot) \\ &= T^{k+1}(\bot) \quad \text{from proposition 10} \end{aligned}$$

Then, it is easy to prove that
$K \uparrow \omega = T \uparrow \omega$.

Proposition 14 $K \uparrow \omega = T \uparrow \omega$
Proof

$$\begin{aligned} &K \uparrow \omega \\ &= K^\omega(\bot) \\ &= lub\{K^n(\bot) \mid n \geq 0\} \\ &= lub\{T^n(\bot) \mid n \geq 0\} \quad \text{from proposition 13} \\ &= T^\omega(\bot) \\ &= T \uparrow \omega \end{aligned}$$

Therefore we get the following theorem.

Theorem 5

$$lfp(K) = K \uparrow \omega = T \uparrow \omega = lfp(T)$$

Proof Obvious.

3 Least Fixpoint Semantics

3.1 Least Fixpoint Semantics

In this section we give the least fixpoint semantics of logic programs on specialization systems. Let P be a logic program on a specialization system. First we define a *one-step-inference transformation* T_P which gives the result of one step inference using a program P, and we prove T_P is monotonic and lower continuous. Second we define a *knowledge-increasing transformation* $K_P = T_P + I_d$ which represents the increase of knowledge by one step inference using a program P. By using the results in section 2, we conclude that K_P has the least fixpoint $lfp(K_P)$ and it is equal to $K_P \uparrow \omega$.

3.2 T_P and K_P

Proposition 15 $X = powerset(\mathcal{G})$ is a complete lattice with the inclusion relation as a partial order, whose minimal element $\perp$ is the empty set ϕ.
Proof Clear from the definitions.

Definition 16 Let P be a logic program on a specialization system $\Gamma =< \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu >$. For any P we define a transformation T_P on the power set X of $\mathcal{G}$ as follows. For any x in $X = powerset(\mathcal{G})$, we define

$$T_P(x) = \{g \mid \exists H, \exists B_1, \ldots, \exists B_q, \exists\theta : Cond(g, P, H, B_1, \ldots, B_q, \mathcal{S}, \theta, x)\}$$

where, $Cond(g, P, H, B_1, \ldots, B_q, \mathcal{S}, \theta, x)$ is the conjunction of

1. $H \leftarrow B_1, \ldots, B_q$ is a P's program clause.
2. $\theta \in \mathcal{S}$
3. θ is applicable to H and $B_1, \ldots, B_q$.
4. $H\theta, B_1\theta, \ldots, B_q\theta$ are in $\mathcal{G}$.
5. $g = H\theta$
6. $B_1\theta, \ldots, B_q\theta$ are elements in x.

T_P is called the *one-step-inference transformation* of P.

In the following we use $Cond(g, x)$ as an abbreviation of the above defined condition, $Cond(g, P, H, B_1, \ldots, B_q, \mathcal{S}, \theta, x)$.

Proposition 16 $T_P : X \to X$ is monotonic.
Proof Clear from the form of the condition 6 in $Cond(g, x)$.

Proposition 17 $T_P : X \to X$ is lower continuous.
Proof It is enough to prove

$$g \in T_P(lub\{x_n\}) \Leftrightarrow g \in lub\{T_P(x_n)\}$$

for any monotonically increasing sequence $\{x_n\}(n \in N)$ on X. From the assumption that $\{x_n\}(n \in N)$ is a monotonically increasing sequence and that T_P is monotonic, $\{T_P(x_n)\}(n \in N)$ is a monotonically increasing sequence. Then we have:

$$\begin{aligned} & g \in T_P(lub\{x_n\}) \\ & \Leftrightarrow \exists H, \exists B_1, \ldots, \exists B_q, \exists\theta : Cond(g, lub\{x_n\}) \\ & \Leftrightarrow \exists H, \exists B_1, \ldots, \exists B_q, \exists\theta, \exists m : Cond(g, x_m) \\ & \Leftrightarrow \exists m : g \in T_P(x_m) \\ & \Leftrightarrow g \in lub\{T_P(x_n)\} \end{aligned}$$

Therefore, $T_P(lub\{x_n\}) = lub\{T_P(x_n)\}$.

Definition 17 For any logic program P, a transformation K_P on the powerset X of $\mathcal{G}$ is defined by $K_P = T_P + I_d$, where I_d is the identity mapping on X. K_P is called the *knowledge-increasing transformation* of P.

Proposition 18 K_P is increasing, monotonic and lower continuous.
Proof Straightforward from theorem 2 and from propositions 15, 16 and 17.

3.3 Fixpoint Theorem for K_P

We can state that X is a fixpoint of K_P iff we can not increase X by the inference of P. Therefore, we have a good reason why we believe that what the program P defines is one of the fixpoints of K_P.

Theorem 6 $lfp(K_P, x) = {K_P}^{\omega}(x)$ for any x in $powerset(\mathcal{G})$.
Proof Obvious from theorem 1 and proposition 18.

This theorem states that the least elements in the set of all fixpoints of K_P which include x is given by $K^{\omega}(x)$. The usual fixpoint of K_P is a special case of this, because it is identical with the least elements of the set of all fixpoints of K which include $\emptyset$.

Letting x be the empty set, we choose the least fixpoint of K_P as the semantics of P, that is, we regard what P represents as $lfp(K_P)$.

Theorem 7 $lfp(K_P) = K_P \uparrow \omega$
Proof Obvious from theorem 3 and proposition 18.

Assume that $P = \{y \leftarrow x \quad ; \quad z \leftarrow\}$ is a program on the specialization system $< \mathcal{A}_5, \mathcal{G}_5, \mathcal{S}_5, \mu_5 >$ in section 1. Since,

$K_P(\emptyset) = \{p, r\}$
$K_P^2(\emptyset) = K_P(\{p, r\}) = \{p, q, r\}$
$K_P^3(\emptyset) = K_P(\{p, q, r\}) = \{p, q, r, s\}$
$K_P^4(\emptyset) = K_P(\{p, q, r, s\}) = \{p, q, r, s\}$

we can conclude that

$K_P \uparrow \omega = \{p, q, r, s\}$

As it is also clear that

$K_P(\{p, q, r, s, t\}) = \{p, q, r, s, t\}$

fixpoints of K_P are $\{p, q, r, s\}$ and $\{p, q, r, s, t\}$, and the least fixpoint is $\{p, q, r, s\}$. From these,

$$lfp(K_P) = K_P \uparrow \omega = \{p, q, r, s\}.$$

3.4 Fixpoint Theorem for T_P

We can also have the following theorems.

Theorem 8 $lfp(T_P) = T_P \uparrow \omega$
Proof Obvious from theorem 4 and proposition 16 and 17.
Theorem 9

$$lfp(K_P) = K_P \uparrow \omega = T_P \uparrow \omega = lfp(T_P)$$

Proof Obvious from theorem 5.

In spite of the theorem 9, we should not use the fixpoint of T_P in order to define the semantics of P. This is because the fixpoint of T_P has no natural interpretation for explaining the semantics of P. As we stated, the fixpoints of K_P is meaningful because X is a fixpoint of K_P iff we can not increase X by the inference of P. But fixpoints of T_P do not have such good interpretations and it can not provide good reasons to regard the semantics of P as the fixpoint of T_P.

4 Concluding Remarks

The following are the characteristics of the theory in this paper.

- The GLP theory enables us to discuss logic programming from a new abstract level.
- We seek for the most general conditions for the propositions and the theorems, therefore they are very general and widely applicable.
- We do not assume any groundness conditions for interpretation domains.
- We use lower continuous functions rather than continuous functions used in the usual theory. Lower continuity, which is more general than continuity, is enough to develop the least fixpoint theory here.
- The requirement for K is increasing, monotonic and lower continuous, which is more general and easier to understand than the one of usual theory.

- We use K_P instead of T_P as the mapping of the fixpoint theorem. This is because there is no good interpretation for the fixpoint of T_P.

Instead of the fixpoint of one-step-inference transformation T_P in the usual theory we have used the the fixpoints of knowledge increasing transformation $K_P = T_P + I_d$. This results in a general and elegant theory of the least fixpoint semantics. K_P also leads us to the elegant theorem: model = fixpoint, which is discussed in [2, 5].

References

[1] Ait-Kaci,H. and Nasr,R. : LOGIN : A Logic programming language with built-in inheritance, *J. of Logic Programming*, Vol.3 No.3, pp.185-215 (1986)

[2] Akama,K. : Declarative Semantics of Logic Programs on Parameterized Representation Systems, *Hokkaido University Information Engineering Technical Report*, HIER-LI-9001 (1990)

[3] Akama,K. : Generalized Logic Programs on Specialization Systems, *Preprints Work. Gr.for Programming, IPSJ* 6-4-PRG, pp.31-40 (1992)

[4] Akama,K. : Minimal Model Semantics of Generalized Logic Programs, *Preprints Work. Gr.for Programming, IPSJ* 7-7-PRG, pp.49-56 (1992)

[5] Akama,K. : Declarative Semantics of Generalized Logic Programs, (1992 to appear)

[6] Apt,K.R. and van Emden,M,H. : *Contribution to The Theory of Logic Programming, J.ACM*, Vol.29, No.3 (1982)

[7] Clark,K.L. : Logic Programming Schemes, *Proceedings of the international conference on fifth generation computer systems* (1988)

[8] Jaffar,J.,Lassez,J.L. and Maher,M.J : Logic Programming Language Scheme, *Logic programming, functions, relations, and equations*, pp.441-467, Prentice Hall (1986)

[9] Jaffar,J. and Lassez,J.L. : Constraint Logic Programming, *Technical Report, Department of Computer Science, Monash University*, June (1986)

[10] Lloyd,J.W. : Foundations of Logic Programming, 2nd edition, Springer-Verlag, p.212 (1987)

[11] Robinson,J.A. : A Machine-oriented Logic Based on the Resolution Principle, *J.ACM*, vol.12, No.1, pp.23-41 (1965)

[12] van Emden,M.H. and Kowalsky,R.A. : The semantics of Predicate logic as a Programming Language, *J.ACM*, vol.23, No.4 pp.733-742 (1976)

Appendix

1. **Proof for lemma 1**
   Assume that $x \le y$ and $z \le w$, then

$$\begin{array}{ccccc} x & \le & y & \le & y+w \\ z & \le & w & \le & y+w \end{array}$$

   Therefore, $y + w$ is an upper bound of $\{x, z\}$. From the definition, $x + z$ is the least upper bound of $\{x, z\}$. Then we get $x + z \le y + w$.

2. **Proof for lemma 2**

   (a) The proof of $lub\{x_n\} + lub\{y_n\} \le lub\{x_n + y_n\}$.
   $\forall n : x_n \le x_n + y_n \le lub\{x_n + y_n\}$
   $\forall n : y_n \le x_n + y_n \le lub\{x_n + y_n\}$
   Then $lub\{x_n+y_n\}$ is an upper bound of $\{x_n\}$ and $\{y_n\}$, then

$$\begin{array}{ccc} lub\{x_n\} & \le & lub\{x_n + y_n\} \\ lub\{y_n\} & \le & lub\{x_n + y_n\} \end{array}$$

   From lemma 1,

$$lub\{x_n\} + lub\{y_n\} \le lub\{x_n + y_n\}$$

(b) The proof of $lub\{x_n\} + lub\{y_n\} \geq lub\{x_n + y_n\}$. From the definition of lowest upper bound,

$$\forall n: \ lub\{x_n\} \geq x_n$$
$$\forall n: \ lub\{y_n\} \geq y_n$$

From lemma 1,

$$\forall n: lub\{x_n\} + lub\{y_n\} \geq x_n + y_n.$$

Thus $lub\{x_n\}+lub\{y_n\}$ is an upper bound of $\{x_n + y_n\}$, then

$$lub\{x_n\} + lub\{y_n\} \geq lub\{x_n + y_n\}.$$

3. **Proof for proposition 7**
Letting $T = T_1 + T_2$ we prove the monotonicity of T. Assume that $x \leq y$. As T_1 and T_2 are monotonic,

$$T_1(x) \leq T_1(y)$$
$$T_2(x) \leq T_2(y)$$

Then from lemma 1,

$$T_1(x) + T_2(x) \leq T_1(y) + T_2(y)$$
$$T(x) \leq T(y)$$

4. **Proof of proposition 8**
Letting $T = T_1 + T_2$, we prove that T is lower continuous. Let $\{x_n\}(n \in N)$ be any increasing sequence on L. Since T_1 and T_2 are lower continuous, then

$$T_1(lub\{x_n\}) = lub\{T_1(x_n)\}$$
$$T_2(lub\{x_n\}) = lub\{T_2(x_n)\}$$

Adding each side of the formula and using from lemma 2,

$$T(lub\{x_n\}) = lub\{T(x_n)\}.$$

5. **Proof for lemma 3**

(a) The proof of monotonicity. Assume $x \leq y$, then $I_d(x) = x \leq y = I_d(y)$. This leads to $I_d(x) \leq I_d(y)$.

(b) The proof of lower continuity. From the definition of I_d, $I_d(lub\{x_n\}) = lub\{x_n\}$ and $lub\{I_d(x_n)\} = lub\{x_n\}$, then $I_d(lub\{x_n\}) = lub\{I_d(x_n)\}$.