ICOT Technical Report: TR-694

TR-694

INFERENCE TRANSFORMATION
– A New Methodology for the Program Transformation –

by
J. Yamaguchi (Kanagawa Univ.)

September, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

I N F E R E N C E T R A N S F O R M A T I O N

-a New Methodology for the Program Transformation-

by

Jinsei Yamaguchi

Dept. of Information and Computer Sciences
KANAGAWA UNIVERSITY
2946 Tsuchiya, Hiratsuka
Kanagawa 259-12 JAPAN

Abstract: We define the concept of the "inference transformation" as a completely new methodology for the program transformation in the field of the logic programming, and we show the theoretical background.

Key Words: logic, AI, inference

§0. Introduction

A program transformation would have two purposes which are related to each other. One is to gain an efficiency and the other is to generate a variety of programs from the original one. In anyway, the original and the transformed should have a certain kind of common property because we can't use the word "transformation" if there exists no relation between them at all. Talking the case of the logic programming, the common property is embodied by the preservation of the intended model. In other words, when we regard a logic program as a formal theory, the set of ground atoms proved by the original P and the set of ground atoms proved by the transformed P' should coincide.
($\{\sigma \in Ap \mid P \vdash \sigma\} = \{\sigma \in Ap \mid P' \vdash \sigma\}$, where Ap is the set of all ground atoms generated by the language which is used to describe P.)

Now, we can regard the execution of a pure logic program as applying "a set of inference rules" to a given "set of axioms". (For example, SLD-resolution + NAF-rule to a set of Horn clauses.) Standing on this viewpoint, we may be able to say that the target of the conventional notion of the (logic) program transformation is solely limited to the case of the change of a set of axioms while preserving the inference rule(s) employed.(From an angle of a formal theory, the change of, say, the backward and the forward reasonings are trivial.) Here, if we stand for the viewpoint that a theory is made from the composition of a set of axioms and a set of inference rules where both concepts are inherently related and influenced to each other, there ought to be an approach to a theory transformation in a wider sense that both axioms and inference rules are changed. (As for an extreme case, just consider two formalizations of Hilbert style and Gentzen style of the first order classical logic.) Of course, to establish this new approach theoretically, a variety of inference rules should be supposed to be available. In other words, we must admit the situation where where modus ponens is not the only inference rule employed by

the logic programming.

For example, when an equivalence relation $\equiv$ over the set of all atoms generated from a Horn clause program P is defined by somehow or other, it is natural for us to admit the following form of the inference rule.

$$\frac{A,\ A' \rightarrow B}{B'} :\equiv \quad \text{, where } A \equiv A' \text{ and } B \equiv B'.$$

The above stated inference transformation can be applied to this kind of inference rule as a simple case.

Here, let E be an equational theory (over a set of all terms generated from the language Lp over P) which defines an equivalence relation $\equiv_E$. In this case, we at once notice the techinique of syntactically changing E to another equational theory E' such that $\equiv_E \restriction Bp \times Bp = \equiv_{E'} \restriction Bp \times Bp$, where Bp is the Herbrand base of P. (See, for example [Do].) However, there are many cases where even two different equivalence relations $\equiv$ and $\equiv'$ can satisfy the condition

$$\equiv \restriction Bp \times Bp \neq \equiv' \restriction Bp \times Bp$$

but

$$\{\sigma \in Bp \mid (P, \equiv) \vdash \sigma\} = \{\sigma \in Bp \mid (P, \equiv') \vdash \sigma\} \qquad \cdots(1)$$

In this paper, we would like to systematically consider inference transformations which treat the change from $\equiv$ to $\equiv'$ that preserves both the above conditions (1) within the framework of Boolean-valued logic programming (language) scheme LIFE-Ⅲ.

§1. Preliminaries

In this section, we define basic terminologies and notions which are used in the subsequent sections. We employ Horn clause programs as the target logic programs. Throughout the paper, let P be a Horn clause program and Up(Bp) be the Herbrand universe (base) of the language Lp used in P. Moreover, let $T(\Sigma p, V)$ ($A(\Pi p, V)$) be the set of all terms (atoms) generated from Lp whose variables are picked up from V. Obviously, $Up \subset T(\Sigma p, V)$ and $Bp \subset A(\Pi p, V)$. Here,

Definition 1-1.

i) An equivalence relation $\equiv$ over $A(\Pi p, V)$ is substitution transitive iff
$(\forall A, B \in A(\Pi p, V))\ (\forall \theta$: substitution over $T(\Sigma p, V))\ (A \equiv B \rightarrow A\theta \equiv B\theta)$.

ii) Let $\equiv$ be a substitution transitive equivalence relation over $A(\Pi p, V)$ and $A, B \in A(\Pi p, V)$. Then, a substitution θ over $T(\Sigma p, V)$ is a $\equiv$-unifier for A and B iff
$A\theta \equiv B\theta$. ⊣

Using the above defined notion of $\equiv$-unifier, we can naturally extend the notion of syntactical unification to that of $\equiv$-unification which can be employed as the base of inference rule over P.

Remark: Precisely speaking, we had better define the notion of $\equiv$-mgu here to ensure the realizability of logical completeness as a logic program $(P, \equiv)$. However, without knowing the definition of $\equiv$-mgu, we can continue the following argument. ⊣

In this situation, we obtain the following notion as a subset of Bp.

Definition 1-2.

$Sp(\equiv) = \{\sigma \in Bp \mid P \cup \{\leftarrow \sigma\}$ has a refutation based on $\equiv$-unification$\}$. ⊣

Now, let B be a complete Boolean algebra and F be a complete filter over B. Then, using F, we can induce an equivalence relation $\equiv_F$ over the set of all maps

$$\{f \mid f : Bp \to B\}$$

in the following way.

Definition 1-3.

i) Define a relation $\leqq_F$ over B^{Bp} by

$$(\forall f, g \in B^{Bp})(f \leqq_F g \quad \text{iff} \quad (\forall \sigma \in Bp)((f(\sigma) \to g(\sigma)) \in F)).$$

ii) Define $\equiv_F$ over B^{Bp} by

$$(\forall f, g \in B^{Bp})(f \equiv_F g \quad \text{iff} \quad f \leqq_F g \text{ and } g \leqq_F f). \qquad \dashv$$

Here, define a partially ordered structure $(Pp, \leqslant_F)$ such that $Pp = B^{Bp}/\equiv_F$ where $\leqslant_F$ over Pp is naturally induced from the corresponding relation $\leqq_F$ over B^{Bp}. That is, for any $[f], [g] \in Pp$,

$[f] \leqslant_F [g]$ iff $\quad f \leqq_F g$ for suitable repesentatives $f \in [f]$ and $g \in [g]$.

Obviously, the above $\leqslant_F$ does net depend on the choice of the representatives and so is well-defined. Then, $(Pp, \leqslant_F)$ becomes a complete lattice w.r.t. the operations $\vee, \wedge$ induced pointwisely from the operation on B.

(For example, for any $X \subset Pp$, we can decide

$$\wedge X = [g] \quad \text{so that}$$

$$(\forall \sigma \in Bp)(g(\sigma) = \bigwedge_{[f] \in X} f(\sigma))$$

where f is a representative of [f]. Again, it is easy to check that the definition is well-defined.)

By the way, we can regard any $f \in B^{Bp}$ as a B-valued interpretation of P. In this sense, the following are those in which we are interested from a viewpoint of model theory.

Definition 1-4. Let $f \in B^{Bp}$. Then,

i) (Up, f) is a Herbrand F-model of P iff

$(\forall C \in P)(f(C) \in F)$

where f(C) is the abbreviation of

$$\bigwedge_{\rho:\text{ground}} (f(C^+\rho) \leftarrow f(C^-\rho)).$$ (C^+ is the head of C and C^- is the body of C.)

ii) (P, f) is a F-program iff (Up, f) is a Herbrand F-model of P.

iii) Define a subclass of Pp so that

$Mp(B, F) = \{ [f] \in Pp \mid (P, f) \text{ is a } F\text{-program} \}$. ⊣

Here again, it is obvious that the above $Mp(B, F)$ is independent of the choice of a representative f of [f] and so is well-defined. In addition, it is not so difficult to check that $(Mp(B, F), \leqslant_F)$ becomes a complete sublattice of $(Pp, \leqslant_F)$. Lastly, we define the following subset of Bp.

Definition 1-5. Let $f \in B^{Bp}$. Define

$B_F[f] = \{\sigma \in Bp \mid f(\sigma) \in F\}$. ⊣

By definition, it is obvious that

$(\forall [f] \in Pp)(\forall f, g \in [f])(B_F[f] = B_F[g])$.

§2. Boolean-valued Unification

In this section, we define the notion of "Boolean-valued unification" which is employed by LIFE-III and discuss the related topics including Boolean-valued completeness.

Definition 2-1. Let $\sim$ be a substitution transitive relation over $A(\Pi p, V)$ and $f \in B^{Bp}$. Let $A, B \in A(\Pi p, V)$. Then, a substitution θ over $T(\Sigma p, V)$ is a B-valued unifier for A and B w.r.t. $\sim$ and f iff

1. $A\theta \sim B\theta$

   and

2. $f(A\theta\rho) = f(B\theta\rho)$ for any ground substitution ρ for $A\theta$ and $B\theta$ ⊣

Remark: The second condition above is equal to $\bigwedge_{\rho:\text{ground}} (f(A\theta\rho) \leftrightarrow f(B\theta\rho)) = 1$ ⊣

As is obvious from the definition, the notion of B-valued unification is far more complex than that of universal unification. However, it is this intricacy that provides us a variety of benefits. For example, by employing B-valued unification instead of usual $\equiv$-unification as the basis of an inference rule which governs P, we can obtain a subset of Bp in the following way.

Definition 2-2.

$Sp(\sim, f) = \{\sigma \in Bp \mid P \cup \{\leftarrow \sigma\}$ has a refutation based on B-valued unification w.r.t. $\sim$ and $f\}$. ⊣

Compared the above $Sp(\sim, f)$ with $B_F[f]$ of Definition 1-5, we can see that

$$Sp(\sim, f) \subset B_F[f]$$

always holds. (B-valued soundness property)

However, it is easy to check that the equality

$$Sp(\sim, f) = B_F[f]$$

does not always hold for arbitrary $\sim$ and f. From now on, let's investigate a general condition concerning $\sim$ and f which permits the equality. For this purpose, we need the following concept.

Definition 2-3. Let $J:Bp \to Bp$ be an idempotent function ($J^2=J$). Then,

i) Given a relation $\sim$ over $A(\Pi p, V)$, J is $\sim$-consistent iff $(\forall \sigma \in Bp)(\sigma \sim J(\sigma))$.

ii) $f \in B^{Bp}$ is J-faithful iff

$(\forall \sigma \in Bp)((f(\sigma)=f(J(\sigma)))$. ⊣

Now, suppose an idempotent $\sim$-consistent J is given. Then, we can consider a subclass $\Gamma p(J)$ of Pp such that

$\Gamma p(J)=\{[f] \in Pp \mid f \text{ is J-faithful}\}$.

Here, it is easy to see that $(\Gamma p(J), \leqslant_F)$ becomes a complete sublattice of $(Pp, \leqslant_F)$.

Remark: By definition, for any $[f] \in \Gamma p(J)$, there always exists a representative $f \in [f]$ which is J-faithful. However, owing to the flexibility of F, there may be an element $g \in [f]$ which is not J-faithful. ⊣

Example 2-4.

$$\text{Let } P=\begin{cases} p\leftarrow & \\ q\leftarrow r & , \end{cases} \qquad p\sim q\sim r,$$

$J(p)=p, J(q)=q, J(r)=p, B=\{1, 0, b, \neg b\}, F=\{1, b\}$.

Then, $Bp=\{p, q, r\}$. Define $f, g \in B^{Bp}$ so that

$f(p)=f(q)=f(r)=1,\ g(p)=g(q)=1, g(r)=b$.

Then, J is idempotent and $\sim$-consistent, $[f] \in Mp(B, F) \cap \Gamma p(J)$, $[f]=[g]$ and f is J-faithful.

However, it is obvious that g is not J-faithful. ⊣

With the above observations in mind,

Theorem 2-5. Let $\sim$ be a substitution transitive equivalence relation over $A(\Pi p, V)$ and J be a $\sim$-consistent idempotent function. Let $[\mu]$ be the least element of $\Gamma p(J) \cap Mp(B, F)$. Then,

$$Sp(\sim, \mu) = B_F[\mu]$$

where μ is a J-faithful representative of $[\mu]$. □

As a direct consequence, we notice that

Corollary 2-6. Using the notations in Theorem 2-5, let's μ_1, μ_2 be two distinct J-faithful elements of $[\mu]$. Then,

$$Sp(\sim, \mu_1) = Sp(\sim, \mu_2).$$ □

By the way, during the proof of Theorem 2-5, we use the following lemma, which will be used later.

Lemma 2-7. Using the notations in Theorem 2-5, $[\mu]$ always has a J-faithful representative μ_2 such that

$$(\forall \sigma \in Bp)\ (\mu_2(\sigma) = 1 \text{ or } \mu_2(\sigma) = 0).$$ □

For the proofs of the above results, see [8], [9].

§3. Theoretical Background of Inference Transformation

In this section, we consider the theoretical aspects of inference transformation. To begin with, in general, there are many methods (at least theoretically) which substantially embody the same effects as a given $\equiv$-unification by the tool of B-unification using suitable $\sim$ and $f \in B^{Bp}$. As for the simplest, just consider the case

$\equiv = \sim$ and $f = T$, i.e., $(\forall \sigma \in Bp)(f(\sigma) = 1)$.

Now, suppose we can choose $\sim$ and f such that

$(\forall \sigma, \tau \in Bp)(\sigma \equiv \tau$ iff $\sigma \sim \tau$ and $f(\sigma) = f(\tau))$.

Then, our aim is to choose a representative $g \in [f]$ such that

$$g \neq f \text{ and } Sp(\sim, f) = Sp(\sim, g) \qquad \cdots (2)$$

through the flexibility of F. However, there is no assurance that (2) holds for any candidate g of $[f]$.

Here is the place where Theorem 2-5 in the previous section essentially works. The main purpose of this paper is the following result.

Theorem 3-1. Soppose a substitution transitive equivalence relation $\equiv$ over $A(\Pi p, V)$ is given. Let $\sim, J, B, F, [\mu]$ be such that

1) B is an arbitrary complete Boolean algebra and F is an arbitrary complete filter over B.
2) $\sim$ is a substitution transitive equivalence relation over $A(\Pi p, V)$ which satisfies $\sim \restriction Bp \times Bp \supseteq \equiv \restriction Bp \times Bp$
3) J is a choice function of $\equiv$, i.e.,
   $(\forall \sigma, \tau \in Bp)(\sigma \equiv \tau$ iff $J(\sigma) = J(\tau))$
   (In this case, J obviously becomes $\sim$-consistent and idempotent.)
4) $[\mu]$ is the least element of $\Gamma p(J) \cap Mp(B, F)$.

Then, $Sp(\equiv) = Sp(\sim, \mu)$

for any J-faithful representative μ of $[\mu]$.

Proof: Let $\sim$ and J be as in 2) and 3). Let B' and F' be complete Boolean algebra and complete filter over B' such that

$$|Bp/\equiv| \leqq |F'| .$$

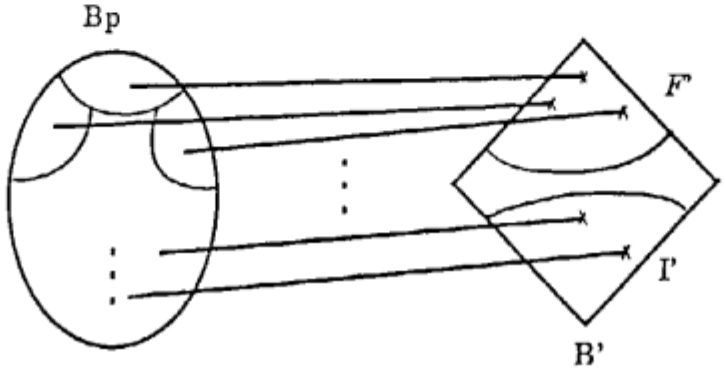
Consider the least element $[\mu']$ of $\Gamma p(J) \cap Mp(B', F')$. Here, pick up a J-faithful representative μ_J of $[\mu']$ such that

$$(\forall [\sigma], [\tau] \in Bp/\equiv)\,([\sigma] \neq [\tau] \leftrightarrow \mu_J(\sigma) \neq \mu_J(\tau)).$$

That is,

$$(\forall \sigma, \tau \in Bp)\,(\sigma \equiv \tau \quad \text{iff} \quad \mu_J(\sigma) = \mu_J(\tau)).$$

(By the choice of B', F', this sort of selection of μ_J from $[\mu']$ is always possible.)



Using this μ_J, we notice that

$$(\forall \sigma, \tau \in Bp)\,(\sigma \equiv \tau \quad \text{iff} \quad \sigma \sim \tau \text{ and } \mu_J(\sigma) = \mu_J(\tau))$$

$\Rightarrow$ θ is a (ground) $\equiv$-unifier for A and B

iff

θ is a (ground) B'-valued unifier for A and B w.r.t. $\sim$ and μ_J

$\Rightarrow$ $Sp(\equiv) = Sp(\sim, \mu_J)$.

So, by especially choosing $\mu'_2 \in [\mu']$ such that

$$(\forall \sigma \in Bp)\,(\mu'_2(\sigma) = 1 \text{ or } \mu'_2(\sigma) = 0),$$

we obtain

$$Sp(\equiv) = Sp(\sim, \mu'_2).$$

Now, for any complete Boolean algebra B and a complete filter F over B, we can consider the least element $[\mu]$ of

$$\Gamma p(J) \cap Mp(B, F).$$

Here again, we can choose a J-faithful representative μ_2 of $[\mu]$ such that

$(\forall \sigma \in Bp)(\mu_2(\sigma)=1 \text{ or } \mu_2(\sigma)=0)$.

Since both (Up, μ_2) and (Up, μ'_2) are the least J-faithful, 2-valued {1}-model of P, $\mu_2=\mu'_2$.

So, for any J-faithful representative μ of $[\mu]$, we get

$Sp(\equiv)=Sp(\sim, \mu)$.

(In this proof, we use the result of Lemma 2-7.) □

The necessity of conditions 3) and 4) of the above result can be guessed by the next example.

Example 3-2. Let P, B, F, $\sim$, f, g be as in Example 2-4.

i) Suppose $p \equiv q \equiv r$.

Let J be as in Example 2-4.

Then, $Sp(\equiv)=Sp(\sim, f)=B_F[f]=\{p, q, r\}$

$\neq \{p, q\}=Sp(\sim, g)$.

In this example, [f] is the least element of $Mp(B, F)\cap\Gamma p(J)$ and $g\in[f]$ but g is not J-faithful.

ii) Suppose $p \equiv q$.

Let J' be such that $J'(p)=p$, $J'(q)=p$, $J'(r)=r$.

Then, $Sp(\equiv)=Sp(\sim, g)=\{p, q\}$

$\neq Sp(\sim, f)=B_F[f]=\{p, q, r\}$

In this example, both f and g is J'-faithful and $[f]=[g]$, but [f] is not the least element of $Mp(B, F)\cap\Gamma p(J')$.

iii) Suppose $p \equiv q \equiv r$.

Let J" be such that $J''(p)=p$, $J''(q)=q$, $J''(r)=q$ and $h\in B^{Bp}$ be such that $h(p)=1$, $h(q)=h(r)=0$.

Then, [h] is the least element of $Mp(B, F)\cap\Gamma p(J'')$ and h is J"-faithful.

Moreover, J" is $\sim$-consistent.

However,

$Sp(\sim, h) = \{p\} \neq \{p, q, r\} = Sp(\equiv)$.

This is because J" is not a choice function of $\equiv$. ⊣

iv) Suppose $q \equiv r$.

Change $\sim$ to $\sim'$ such that $q \sim' r$. Let J be as in i) and $k \in B^{Bp}$ be such that $k(p) = k(r) = 1, k(q) = 0$.

Then, $Sp(\equiv) = Sp(\sim', f) = Sp(\sim', k) = \{p\}$

and f, k is J-faithful and [f] is the least element of $Mp(B, F) \cap \Gamma p(J)$. However, $[k] <_F [f]$. This is because $[k] \notin Mp(B, F)$. In addition,

$B_F[f] = \{p, q, r\} \neq Sp(\sim', f)$.

This deficiency comes from the fact that J is not $\sim'$-consistent. ⊣

By combining the above theorem 3-1 and Corollary 2-6, we can obtain the expected theory of inference transformation. Here again, by using Example 2-4, let's grasp the idea of inference transformation in its simplest form.

Example 3-3. Let P be as in Example 2-4. Suppose $\equiv$ is defined so that $p \equiv r$. Then, obviously, B, F, $\sim$, J in Example 2-4 satisfies the conditions of Theorem 3-1. Let [μ] be the least element of $\Gamma p(J) \cap Mp(B, F)$. Then, f in Example 2-4 becomes a J-faithful representative of [μ]. (The check is easy.)

In this situation, let μ, μ' ∈ [μ] be such that

$\mu(p) = \mu(r) = 1, \mu(q) = b$

and

$\mu'(p) = \mu'(r) = b, \mu'(q) = 1$.

Then, by definition, both μ and μ' are J-faithful. Here, note the fact that it is μ (but not f) that embodies the given equivalence relation $\equiv$ in the sense that

$(\forall \sigma, \tau \in Bp)(\sigma \equiv \tau$ iff $\sigma \sim \tau$ and $\mu(\sigma) = \mu(\tau))$.

Now, the inference rule based on B-valued unification w.r.t. ($\sim$, μ) and the one w.r.t. ($\sim$, f) become different. To see this, just check the query {←q}. Though ←q and p← can't be B-valued unified w.r.t. ($\sim$, μ), ←q and p← are B-valued unifiable

w.r.t. (~, f). In this sense (of reducing the number of input clauses used), we gain an efficiency by inference transformation from μ to f. A remarkable fact is that this optimization is absolute in the sense that it does not depend on the character of the given query. As far as Bp(={p, q, r}) concerns, for any query {←p}, {←q}, {←r}, using (~, f) is always more efficient than using (~, μ).

On the other hand, there is a case of inference transformation whose optimization entirely depends on each particular query.

As an end to this direction, we enter the area of B-valued reasoning where the evaluation of B-value at each unification step essentially works. In this example, compare inference rules based on B-valued unification w.r.t. (~, μ) and (~, μ'). There is no difference between them except that B-values estimated by μ during B-valued inference becomes different from the one estimated by μ'. An argument concerning the availability of B-values will appear soon, too.(See [13]) ⊣

§4. B-valued Interpretation of ≡-unification

In this section, we consider the embodyment of ~, J, B, F which are used in Theorem 3-1. This becomes crucial when the result is applied to a practical phase. First of all, by reviewing the result of Theorem 3-1 carefully, we should notice that, owing to the general property of the claim stated, there is a possibility that [μ] can't witness the given ≡ in the sense that

$(\exists \mu \in [\mu])(\mu \text{ is } J\text{-faithful and } (\forall \sigma, \tau \in Bp)(\sigma \equiv \tau \text{ iff } \sigma \sim \tau \text{ and } \mu(\sigma) = \mu(\tau)))$ ··· (3)

does not hold.

Example 4-1. Let P, ~, J be as in Example 2-4. Suppose $p \equiv r$. Take $B = \{1, 0\}$ and $F = \{1\}$. Then B, F, ~, J satisfy the conditions 1) ~ 4) in Theorem 3-1. Let [μ] be the least element of $\Gamma p(J) \cap Mp(B, F)$. Then, it is easy to see that

$[\mu] = \{f\}$ and f is J-faithful

where f is in Example 2-4. So,

$Sp(\equiv) = Sp(\sim, f) = \{p, q, r\}$

surely holds. However,

$p \sim q$ and $f(p) = f(q)$ but $p \not\equiv q$.

So, this [μ] violates the condition (3). ⊣

However, as far as the notion of inference transformation concerns, the general form of Theorem 3-1 is enough to produce another inference rule from the original one based on ≡. In this general case, the realization of ~, J, B, F becomes the following.

Suppose a substitution transitive equivalence relation ≡ is given.

I. J is nothing but $\equiv \lceil Bp \times Bp$ itself and so ought to be known and uniquely determined from the beginning. Here, note the merit of determination of J instead of ≡ itself. We may consider only $\equiv \lceil Bp \times Bp$.

II. There are many possibilities to decide ~ which satisfies the condition 2) in Theorem 3-1. One possibility is to decide $\sim = \equiv$. In this case, whatever B and F may be, it is not difficult to check that the resulting [μ] always witness ≡ in the sense of (3) by using Lemma 2-7. However, from a practical

viewpoint, we often choose $\sim$ to be trivial in the sense that

$(\forall A, B \in A(\Pi p, V))(A \sim B)$.

In this latter case, we can ignore the rule of $\sim$, but phenomina like Example 4-1 may occur.

III. Since B and F are permitted to be free, as an extreme case, we may decide $B=\{1, 0\}$ and $F=\{1\}$. However, from a viewpoint of inference transformation, we had better choose an appropriate B and F because it is the flexibility of F that permit essentially different candidates of representatives of $[\mu]$. To tell the truth, if $B=\{1, 0\}$ and $F=\{1\}$, then $[\mu]$ always becomes a singleton.

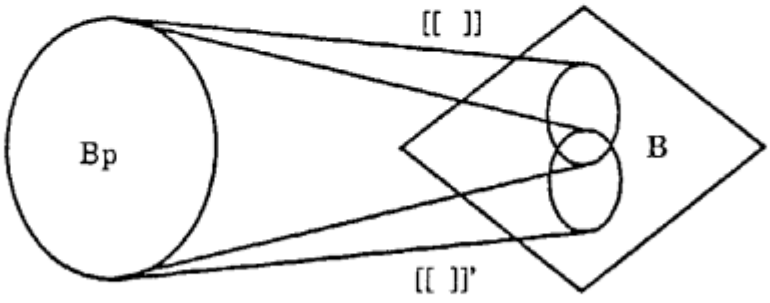
Now, suppose the above $B, F, \sim, J$ is fixed and we choose the least element $[\mu]$ of $\Gamma p(J) \cap Mp(B, F)$ such that the condition (3) is satisfied, where $(\sim, \mu)$ becomes the witness of $\equiv$. Then, is there any intrinsic difference between the original $\equiv$-unification and B-valued unification based on $(\sim, \mu)$? At a first glance, it apparently seems that there is no substantial difference between them, though there really is. It comes from the fact that $[\mu]$ is the least element of $\Gamma p(J) \cap Mp(B, F)$. To explain the situation, let's consider the following more general case. Given $\sim$ and B, we can take two maps

$[[\ ]], [[\ ]]' : Bp \to B$ such that

$(\forall \sigma, \tau \in Bp)([[\sigma]] = [[\tau]]$ iff $[[\sigma]]' = [[\tau]]')$.

As the result, we obtain

$$Sp(\sim, [[\ ]]) = Sp(\sim, [[\ ]]').$$



Once F is fixed, however, whether (P, [[ ]]) becomes a F-program or not is a entirely different issue which concerns B-valued semantics. Standing on this general veiw, we might be able to claim the following. On the one hand, $\equiv$-unification is the concept which does not distinguish the above [[ ]] and [[ ]]'. On the other hand, B-valued unification (w.r.t. $\sim$ and [[ ]]) is the notion which inherently include B-valued semantics in the sense of F-model. It is this intention toward model theory that characterize an advantage of B-valued unification which usual universal unification can't provide.

By the way, in order to enjoy many advantages, we ought to discover the least element $[\mu]$ of $\Gamma p(J) \cap Mp(B, F)$ at the starting point. Here, even comparing the $\equiv$, finding the $[\mu]$ is not so difficult. This fact depends on the following observation. When we try to program (or more generally, formalize) a certain knowledge, we ought to recognize what we are going to program (formalize) from the beginning. This means it is the intended model that comes first. In other words, the procedural semantics (based on $\equiv$-unification or $(\sim, \mu)$-unification) is nothing but one realization of the declarative semantics of the (logic) program. In this situation, our claim is that:

「we had better B-valued interprete the target knowledge so that the above initial $B, F, \sim, J$ are fixed and, at the same time, $[\mu]$ becomes the intended (B-valued) model, instead of simply choosing a naive and primitive $\equiv$.」

" A little refinement ($[\mu]$) of the formalization of the target knowledge (compared with $\equiv$) gives us a great gain."

This is the catch phrase of our B-valued methodology.

§5. Concluding Remarks

In this paper, we focus our attention to one aspect of several features which Boolean-valued logic programming (language) scheme LIFE-Ⅲ owns and apply the character to the area of the inference transformation, the concept of which (a new methodology for the program transformation) is proposed in this paper for the first time. As the result, we provide a theoretical background to a method which transforms the procedural semantics of a logic program P as a whole by changing inference rules that govern P, instead of changing P itself. This approach is our original, we dare claim. (Compare our result with the conventional notion of the "partial evaluation".)

As can be easily guessed, the techinique of the inference transformation is closely related to the machine learning, especially to the topics of chunking, category change, concept discovery etc. Precise arguments will appear soon.

References

[1] Jaffar,J.,Lassez,J.L. and Maher,M.J., "A Logic Programming Language Scheme", in: D.DeGroot and G.Lindstrom(eds.), Logic Programming:Relations,Functions and Equations,(Prentice Hall 1986),441-467.

[2] Lloyd,J.W., "Foundations of Logic Programming",Springer-Verlag,1984.

[3] Maher,M.J., "Equivalences of Logic Programs", Proc. of the 3rd International Conference on Logic Programmin, 1986, 410-424.

[4] Rosenbloom,P.S. and Newell,A., "The Chunking of Goal Hierarchies", in Machine Learning: An Artificial Intelligence Approach,Vol.Ⅱ, R.S.Michalski, L.G.Carbonell and T.M.Mitchell(eds.),(Morgan Kaufmann 1986).

[5] Siekmann,J. and Szabo,P., "Universal Unification and a Classification of Equational Theories",Proceedings of the 6th Conference on Automated Deduction, in: D.W. Loveland(ed.),Lecture Notes in Computer Science 138(Springer Verlag 1982),369-389.

[6] Stepp,R.E. and Michalski,R.S., "Conceptual Clustering:Inventing Goal-Oriented Classifications of Structured Objects", in Machine Learning: An Artificial Intelligence Approach,Vol.Ⅱ, R.S.Michalski, L.G.Carbonell and T.M.Mitchell(eds.), (Morgan Kaufmann 1986).

[7] Tamaki,H. and Sato,T., "Unfold/Fold Transformation of Logic Programs", Proc. of the 2nd International Logic programming Conference, 1984, 127-138.

[8] Yamaguchi,J., "Boolean-valued Logic Programming Language Paradigm:LIFE-Ω—Theoretical Background of LIFE-Ⅰ,Ⅱ,Ⅲ—",NEC LR-5197,1987, revised version to appear.

[9] Yamaguchi,J., "Logical Completeness of LIFE-Ⅱ and LIFE-Ⅲ and Its Applications to the Foundation of Logic Programming",NEC LR-5346,1987, revised version to appear.

[10] Yamaguchi,J., "Universal Unification from a Viewpoint of LIFE-Ⅱ",NEC LR-5347, 1987, revised version to appear.

[11] Yamaguchi,J., "Boolean-valued Logic Programming Language Scheme:LIFE-Ⅲ [1]Infe-

rentially Transforming Logic Programs",(in Japanese)5th Conference Proceedings Japan Soc.Software Sci.&Tech.,1988,237-240.

[12] Yamaguchi,J., "LIFE-Ⅲ [3]Relativized Completeness", Proc. of the 3rd Annual Conference of JSAI,1989,33-36.

[13] Yamaguchi,J., "LIFE-Ⅲ [4]A Fuzzy Inferential System",(in Japanese) 6th Conference Proceedings Japan Soc.Software Sci.&Tech.,1989,121-124.