TR-695

Boolean-valued Logic Programming Language Scheme LIFE-III
– A Summary –

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

Boolean-valued Logic Programming Language Scheme LIFE-Ⅲ
- A Summary -

by

Jinsei Yamaguchi

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

Abstract: In this paper, we define the notion of LIFE-Ⅲ (Logic-oriented Inferential Framework Extension:type Ⅲ) and sketch the logical background as a Boolean-valued logic programming language scheme. Emphases are put on the nature of both the semantic reasoning and the approximate reasoning on which the scheme is based.

Key Words: Boolean algebra, Boolean-valued logic programming, Approximate reasoning, Semantic unification

§0. Introduction

In [16], we proposed LIFE-Ω (Logic-oriented Inferential Framework Extensions: infinite series) as a Boolean-valued logic programming paradigm and discussed some expected characteristics. The aim of our raising this flag is to treat a variety of topics in AI within (or, at least, in the neighborhood of) the kingdom of logic programming as many as possible. The intention of the aim springs from our belief that any subjects covered by AI should be managed not metaphisically but logically, because they all have the destiny to be related to and so connected with a program which runs on a machine at their final step by somehow or other. Of

course, the word "logically" above must be understood in its widest sense.

In this aspect, however, there is a big barrier laid down in front of us. Even if our belief is right and sound, the logical method we can employ has the restriction in the sense that we can't utilize the full power of the conventional mathematical logic because of the nature. Nevertheless, there still remains a lot of wonderful thoughts, techniques and ideas in logic which we can modify and convert into a well-fitted technology in the world of AI. Our strategy was to partly borrow the *philosophy* of Boolean-valued interpretation of the Forcing Method in set theory in order to enlarge the notion of logical inferences so far exist so that the resulting inference methodology can embody so to speak common sense reasoning, i.e., the semantic reasoning, the approximate reasoning, hypothetical reasoning and the non-monotonic reasoning etc. Thus, we built a completely new logical paradigm which is suitable for AI. (The precise arguments can be seen in [16].)

Once the paradigm had been settled, the next task was to formalize a series of language schemes based on the paradigm. Considering the fashion in that season, we started our show by defining LIFE-O = pure Prolog as a laguage scheme. The successive three entrees LIFE-Ⅰ, LIFE-Ⅱ, LIFE-Ⅲ became straightforward extensions of LIFE-O in the sense that

$$\text{LIFE-O} \subset \text{LIFE-Ⅰ} \subset \text{LIFE-Ⅱ} \subset \text{LIFE-Ⅲ}$$

where $\subset$ is the inclusion among sets of logic programming languages covered by each scheme.

Remark: LIFE-Ⅰ was sometimes called "Boolean-valued Prolog".

Here in this paper, we are going to briefly touch the logical property of LIFE-Ⅲ from a viewpoint of the extended inference methodology. Throughout this paper, we employ conventional terminologies in this field. Thus, for example, P, Up, Bp are exclusively used to denote a definite clause program, the Herbrand universe of P and the Herbrand base of P respectively. $T(\Sigma p, V)$ and $A(\Pi p, V)$ stand for the set of all terms

and the set of all atoms whose variables come from V respectively, etc.

§1. $\mathbb{B}$-valued Deduction of LIFE-Ⅲ

Definition 1-1. Suppose P is arbitrary but fixed. Let $\mathbb{B}$ be a complete Boolean algebra. Let $[\![\ ]\!]: B_P \to \mathbb{B}$ be a map.

1. A reflexive, symmetric relation $\sim$ over $A(\Pi_P, V)$ is called "substitution-monotonic" iff

$(\forall A,B \in A(\Pi_P,V))(\forall \theta$:substitution over $T(\Sigma_P,V))(\ A\sim B \to A\theta \sim B\theta\ )$.

2. Let $\sim$ be a substitution-monotonic relation and $A,B \in A(\Pi_P,V)$. A substitution θ over $T(\Sigma_P,V)$ is called " a $\mathbb{B}$-valued unifier for A and B w.r.t. $[\![\ ]\!]$ and $\sim$ " iff

$A\theta \sim B\theta$
and
$[\![\ A\theta\rho\ ]\!] = [\![\ B\theta\rho\ ]\!]$ for any ground substitution ρ over $T(\Sigma_P,V)$.

On the ground of this definition, we can obtain a suitable notion of $\mathbb{B}$-mgu (maximally general $\mathbb{B}$-valued unifier). The crucial property concerning $\mathbb{B}$-mgu is that, for any $A,B \in A(\Pi_P,V)$ and for any $\mathbb{B}$-valued unifier θ for A and B w.r.t. $[\![\ ]\!]$ and $\sim$,

$\{[\theta'] \mid \theta'$ is a $\mathbb{B}$-mgu for A and B such that $(\exists \rho$:substitution) $(\theta = \theta'\rho)\ \}$

is finite and not empty, where $[\ ]$ is the equivalence class based on renaming substitutions. (See [17] for the precise arguments.) Using the notion of $\mathbb{B}$-mgu, we can define a deduction methodology of a logic programming paradigm by following the SLD-resolution-like technique except that

1. employ $\mathbb{B}$-mgu

and

2. calculate Boolean-value $\bigwedge_{\rho:\text{ground}} [\![A_m \theta_m \rho]\!]$

at each unification step, where A_m is the selected atom and θ_m is a $\mathbb{B}$-mgu for $\mathbb{B}$-valued unification.

Now, suppose $C_1, \cdots, C_n$ be the input clauses and $\theta_1, \cdots, \theta_n$ be the $\mathbb{B}$-mgus used in a Boolean-valued refutation R of $P \cup \{G\}$. Then, the above condition 2 can compute both the resulting "refutation value"

$$v(R) = \left(\bigwedge_{\rho:\text{ground}} [\![C_1 \theta_1 \rho]\!] \right) \wedge \cdots \wedge \left(\bigwedge_{\rho:\text{ground}} [\![C_n \theta_n \rho]\!] \right)$$

and the "answer value"

$$v(\theta) = \bigwedge_{\rho:\text{ground}} ([\![A_1 \theta \rho]\!] \wedge \cdots \wedge [\![A_k \theta \rho]\!])$$

in addition to the answer substitution $\theta = \theta_1 \cdots \theta_n$, where $G = \leftarrow A_1, \cdots, A_k$. This implicitly means we may obtain different refutation values $v(R)$ *and* $v(R')$ for the same goal G (and possibly even for the same answer substitution θ | G) by passing the different branches of a refutation tree.

On the basis of the above defined deduction methodology, we can define a procedural semantics of the logic programming paradigm by fixing concrete $\sim$ and $[\![\ ]\!]$. Especially, we obtain the notion of

Definition 1-2.

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

$Sp([\![\ ]\!], \sim)$ is the " $\mathbb{B}$-valued success set " of P in the sense of $\mathbb{B}$-valued unification w.r.t. $[\![\ ]\!]$ and $\sim$.

LIFE-Ⅲ is the logic programming scheme, each element of which has thus defined procedural semantics in a pure logic programming level. This means, by suitably defining and combining $\sim$ and $[\![\ ]\!]$, we can obtain a var-

iety of techniques of the inference which has never been appeared before in the following sense.

It not only embodies an approximate reasoning in a wider sense such that

$$\frac{A',\ A \to B}{B'} : A \approx A',\ B \approx B'$$

where $\approx$ is not necessarily an equivalence relation, but also realizes a semantic reasoning in its real sense that it emploies the map $[\![\ ]\!]$ which is a Boolean-valued interpretation of the program P. Considering the fact that two main topics in AI are to embody the approximate reasoning and to represent semantics of the target, the merit of our scheme should be obvious. (Precise argument concerning this aspect of our Boolean-valued deduction can be seen in [24],[25].)

§2. $\mathbb{B}$-valued F-model

Since Boolean-valued deduction methodology employed by LIFE-Ⅲ is far more general than the usual inference rules considered in the conventional mathematical logic (especially, in proof theory), we can not utilize the famous and convenient result of the completeness of the first order predicate calculas as the logical background of LIFE-Ⅲ directly. As the matter of fact, to tell the truth, the program $(P, \sim, [\![\ ]\!])$ is no more the first order theory as far as the domain of P concerns because of the existance of the map $[\![\ ]\!]$.

Remark: We ought to take the essentially different viewpoint whenever we consider the whole program $(P, \sim, [\![\ ]\!])$. As a set of axioms, similar to the case of Boolean-valued interpretation in set theory, it might be seen to be the first order at a first glance. However, as a theory, it is neither the first order nor the mere second order because we employ the extended inference rule as above. It has, in a sense, the

"semantic order", the terminology of which may unveil the originality and the novelty of our concept of Boolean-valued deduction.

If the matter is so, then it may be natural for us to worry about the recursiveness of the realization of $(P, \sim, [\![\ ]\!])$. In this respect, since we already have the plan to put fuzzy reasoning and/or even neural network into use to embody $\sim$, our main concern is pointed not to the recursiveness but to the tractability of each concrete reaization. This is our official answer to the above question.

Nevertheless, by enlarging the notion of logical consequence to the notion of Boolean-valued F-model, we can obtain the declarative semantics of LIFE-Ⅲ as the logic programming language scheme and, to our surprise, we can even show the generalized completeness result which claims the coincidence of the procedural semantics based on Boolean-valued deduction defined in the previous section and the declarative semantics defined below.

Let F be a complete filter over $\mathbb{B}$. Our intention is that F determines the boundary inside $\mathbb{B}$ so that any value b belonging to F is taken to be true to the extent of b in the sense of $\mathbb{B}$. Thus, the dual ideal I determines the area of false values. So, there might be some values in $\mathbb{B}$ which is neither true nor false in the above sense. This is the crucial difference between usual ($\mathbb{2}$-valued) interpretation and our $\mathbb{B}$-valued interpretation. With this difference in mind, we can carefully generalize the notion of usual Herbrand model so that, for any $[\![\ ]\!]: Bp \to \mathbb{B}$,

Definition 2-1.

$(Up, [\![\ ]\!])$ is a Herbrand F-model of P
iff
$$(\forall C \in P)\Big(\bigwedge_{\rho:\text{ground}} ([\![C^+\rho]\!] \leftarrow [\![C^-\rho]\!]) \in F\Big),$$
where
$$[\![C^-\rho]\!] = \begin{cases} [\![A_1\rho]\!] \wedge \cdots \wedge [\![A_k\rho]\!], & \text{if C has the form } C^+ \leftarrow A_1, \cdots, A_k. \\ \varnothing, & \text{if C is an unit clause.} \end{cases}$$

Now, for any Herbrand F-model (Up, f) of P, we can define the following subset of Bp.

Definition 2-2. Let (Up, f) be a Herbrand F-model of P. Then,

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

($B_F[f]$ is the set of F-true ground atoms under the interpretation (Up, f).)

Since f represents $\mathbb{B}$-valued interpretation of P, the above notion $B_F[f]$ represents " $\mathbb{B}$-valued F-declarative semantics " of the whole program $(P, \sim, f)$.

Remark: As stated above, since $(P, \sim, f)$ has the semantic-order, we ought to newly define what $(P, \sim, f)$ means as a formal theory. The meaning becomes $\mathbb{B}$-valued F-declarative semantics when (Up, f) is a Hebrand F-model of P. Here, an interesting fact is that the notion of $B_F[f]$ does not directly depend on $\sim$. The reason is that the chice of f itself implicitly depends on both P and $\sim$. So, we had better denote $f_P(\sim)$ instead of f in $B_F[f]$, though we use the abbreviation in the following.

So far, we have gained two subsets of Bp, that is, $Sp([\![\ ]\!], \sim)$ and $B_F[f]$. As for the first relation between these two,

Theorem 2-3. Let (Up, f) be an arbitrary Herbrand F-model of P and $\sim$ be an arbitrary substitution-monotonic relation over $A(\Pi p, V)$. Then,

$$Sp(f, \sim) \subseteq B_F[f].$$

(Soundness of LIFE-Ⅲ. See [17].) □

Here, the important fact is that the above f need not depend on $\sim$. However, the converse direction $Sp(f,\sim) \supseteq B_F[f]$ does not always hold. As stated above, usually, there should be some connection of f with $\sim$ in order to hold the direction.

§3. $\mathbb{B}$-valued F-completeness

Are there any $\sim$ and f such that

$$Sp(f,\sim) = B_F[f] \qquad \cdots \langle 1 \rangle$$

where (Up, f) is a Herbrand F-model of P ? The answer is positive. The method of our proving the existence of such $\sim$ and f is similar to the original technique in the manner that we use the fixedpoint semantics. The rough idea is that, instead of starting from the whole complete lattice $\mathbb{B}^{Bp}/F$, we begin our argument with a complete sublattice $\mathbb{N}p$ of $\mathbb{B}^{Bp}/F$ and obtain the least element $[\mu_N]$ of $\mathbb{N}p$, a representative μ_N of which is expected to become the answer of $\langle 1 \rangle$.

Remark: Note that we can use the equivalence classes $\mathbb{B}^{Bp}/F$ modulo F instead of $\mathbb{B}^{Bp}$, by the definition of F-model. Thus, we obtain the flexibility of the chice of the representatives of the answer class. The merit of this phenominon is discussed in [18].

Since we should clear the requirement of the fixedpoint semantics, not all complete sublattices of $\mathbb{B}^{Bp}/F$ provide the answer. (The map $Tp(\mathbb{N}p): \mathbb{N}p \to \mathbb{N}p$ which is used to show the fixedpoint semantics depends on $\sim$, too.) However, there are many useful complete sublattices of $\mathbb{B}^{Bp}/F$ which pass the test. In [17], we employ a systematic approach to find many sublattices, each of which gives the different answer class. Among them are those which are abstracted by the terminology "J-faithfulness "of $[f] \in \mathbb{B}^{Bp}/F$, where $J : Bp \to Bp$ is an idempotent function. In [17], we showed that any J-faithful least Herbrand F-model (Up, μ_J) becomes the answer of $\langle 1 \rangle$, where J should satisfy a

certain condition concerning $\sim$. The point is that J can vary from one example to another if only it satisfies the initial condition.

§4. Conclusion

There is no necessity of fixing our attention only to the syntactical or even an universal (which is based on an equational theory) unification over a given logic program P. There ought to exist many other rich and fruitful derivation techniques over the same P, and how to realize this kind of extended inference in the form of a program has been a main topic in AI for a long time. In this paper, we have briefly sketched the deduction power of Boolean-valued logic programming language scheme LIFE-Ⅲ from a viewpoint of both the approximate reasoning and the semantic reasoning. Since the inference methodology is genuinly enlarged, it is no wonder of our doubting the legitimacy of Boolean-valued deduction employed by LIFE-Ⅲ. It is the new notion of Boolean-valued F-completeness based on the generalized notion of Boolean-valued F-model that assures the legitimacy.

Some groups in AI may stand for the position that, during the investigation of a practical AI, it is impossible to preserve, or more weakly we need not stick to, such a logically rigid concept like completeness or even consistency of the program. However, if we can assure the consistency or the completeness of the same knowledge through a different and more natural programming methodology, we should move to this new side. In this sense, it is better for us to logically cover any topics in AI as wide as possible by employing more and more newly-appeared logical technologies. This is the background philosophy of our proposing Boolean-valued logic programming paradigm LIFE-Ω. Do in AI as *neo*-logics do !

References

[1] Apt,K.R. and van Emden,M.H., Contributions to the Theory of Logic Programming, *J.ACM.* 29: 841-863 (1982).

[2] Bowen,K.A. and Kowalski,R.A.,Amalgamating Language and Meta Language in Logic Programming, in: K.L.Clark and S.A.Tarnlund(eds.), *Logic Programming*, Academic, New York,1982.

[3] Dincbas,M. and van Hentenryck,P., Extended Unification Algorithms for the Integration of Functional Programming into Logic Programming, *J. Logic Programming*, 4: 199-227 (1987).

[4] van Emden,M.H. and Kowalski,R.A.,The Semantics of Predicate Logic as a Programming Language, *J.ACM.* 23: 733-742 (1976).

[5] Fages,F. and Huet,G., Complete Sets of Unifiers and Matchers in Equational Theories, *Theoretical Computer Science 43*: 189-200 (1986).

[6] Fitting,M., A Kripke-Kleene Semantics for Logic Programs, *J. Logic Programming*. 2: 295-312 (1985).

[7] Ginsberg,M.I., Multi-valued Logics, *Proc. AAAI-86:* 243-247 (1986).

[8] Jaffar,J., Lassez,J.L. and Maher M.J., Logic Programming Language Scheme, in: D.DeGroot and G.Lindstrom (eds.), *Logic Programming: Relations, Functions and Equations*, Prentice-Hall,1985.

[9] Jech,T.J., *Set Theory*, Academic, 1978.

[10] Kowalski,R., *Logic for Problem Solving*, North-Holland, 1979.

[11] Lloyd,J.W., *Foundations of Logic Programming*, Springer, 1984.

[12] Millar,D.A. and Nadathur,G., Higher-order Logic Programming, *Proc. of the 3rd ICLP, Lecture Notes in Computer Science 225*, Springer, 1986.

[13] Morishita,S., Numao,M. and Hirose,S., Symbolical Construction of Truth-value Domain for Logic Program, *Proc. of the 4th ICLP*, 533-555 (1987).

[14] Nilsson,N.J., Probabilistic Logic, Artificial Intelligence, 28:71-87 (1986).

[15] Porto,A., Semantic Unification for Knowledge Base Deduction, circulated manuscript, 1986.

[16] Yamaguchi,J., Boolean-valued Logic Programming Language Paradigm:LIFE-Ω—Philosophical Background—, NEC LR-5196,1987, revised version to appear.

[17] Yamaguchi,J., Boolean-valued Logic Programming Language Scheme: LIFE-Ⅲ—A Theoretical Background —, ICOT TR-639, 1991, revised version to appear.

[18] Yamaguchi,J., Boolean-valued Logic Programming Language Scheme:LIFE-Ⅲ [1]Inferentially Transforming Logic Programs,(in Japanese) *5th Conference Proceedings Japan Soc. Software Sci.&Tech.*,237-240 (1988), revised version to appear.

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

[20] Yamaguchi,J., LIFE-Ⅲ[5]From a Viewpoint of Situation Theory (Ⅰ) (in Japanese) *Proc. of the 4th Annual Conference of JSAI*,93-96(1990), revised version to appear.

[21] Yamaguchi,J., LIFE-Ⅲ[5]From a Viewpoint of Situation Theory (Ⅱ) (in Japanese) *7th Conference Proceedings Japan Soc.Software Sci.& Tech.*,369-372 (1990), revised version to appear.

[22] Yamaguchi,J., Phases,Informations and the Permitting Relation— a Boolean-valued Representation of Knowledge in AI—, accepted in *9th International Congress of LMPS*, 1991.

[23] Yamaguchi,J., **B**-valued *F*-completeness of LIFE-Ⅲ, (in Japanese) *LPC'91*,151-160 (1991), English version to appear in Springer Lecture Note Series.

[24] Yamaguchi,J., The Approximate Reasoning in Logic Programming, to appear.

[25] Yamaguchi,J., Boolean-valued Deduction and the role of a Filter, to appear.

[26] Zadeh,L.A., Fuzzy Logic and Approximate Reasoning, Synthese 30, 407-428 (1975).