プロログプログラムの定義域のいくつかの性質

大谷木　重夫
（電子技術総合研究所）

あらすじ

プロログプログラムは肯定と否定の情報の増加に関する半順序をもつスコット領域E上の連続関数と見なすことができる。
プロログプログラムの実データ領域は肯定と否定が丁度互いに補集合となるEの部分領域E_0でプログラムはその開集合上の連続関数となっている。
本論文ではE_0上の下から半連続関数がほんの少しの領域をのぞいてE上の連続関数に拡張されることを述べる。
またE_0の開集合上で連続な関数はE上の連続関数に拡張される。

1. はじめに

プロログプログラムの意味として最初に考えられたものはプロログプログラムがHorn文としてもつ意味である。即ち与えられたHorn文から三段論法の繰返しで得られる論理式の全体をプログラムの本来もつ意味と考えようというものである。次にKowalski, Emden 等により、プロログプログラムのdenotationalな意味をとらえようとする試みがなされた。彼等の考えではプロログプログラムはpositive information の増大という半順序に関して連続な関数であった。そのような試みのもとでは cut のようなプログラム制御文の意味が定められない。
一方プロログのプログラムのインタプリターに対して正確なdenotational semanticsを与えようという試みもN. D. JonesとA. Mycroftによってなされている。
プロログプログラムはpositive information についてのみ述べているのでnegative information について考慮する必要がないように見えるが、sequentialインタプリターのもとではunificationの失敗を判定する操作を必要とし、それは本文に述べるようにpositive information の増大の順序に対し、連続性を失う。本論文ではpositive 及び negative information 双方の増大という順序に関して連続な束となる空間E上でsequentialなプロログの意味が記述できることを述べる。

2 空間E

分解原理による定理証明ではHerbrand modelという論理式の集合解釈が前提となる。
例えば

$$\forall x \ P(x, f(x))$$

という式の成立はHerbrand 領域をHとして集合Pが

$$P \supset \{(x, f(x)) \mid x \in H\}$$

を満たしていることを意味している。

$$\forall x\ P(x, f(x)) \wedge \forall x\ \neg P(g(a), x)$$

が矛盾することは

$$\{(x, f(x)) \mid x \in H\} \cap \{(g(a), x) \mid x \in H\} = \{(g(a), f(g(a)))\} \neq \emptyset$$

であることから判定される。
集合の共通部分をとることが節を unify することに相当するわけである。
定理証明をプログラムと見なすという立場に立つ論理プログラムの意味を考える場合は、このような集合解釈を尊重するべきであろう。さもないと unification という基本演算が、その意味を失なって単なる記号操作になってしまうからである。
それではプロログのようなプログラムは如何なる関数として解釈されるべきであろうか。

$$P(x) \Leftarrow Q(x)$$

なる節を入力節 $P(f(x))$ に適用すると $Q(f(x))$ が得られるから

$$(P(x) \Leftarrow Q(x))(y) = Q(P^{-1}(y \cap \{P(x) \mid x \in H\}))$$

と定める。確かに

$$(P(x) \Leftarrow Q(x))(\{P(f(x)) \mid x \in H\}) = \{Q(f(x)) \mid x \in H\}$$

が得られる。
このようにプロログプログラムは集合関数として解釈することが自然である。

$$\begin{array}{ll} P(f(x)) \Leftarrow Q(x) & \\ P(x) & \text{(I)} \end{array}$$

というプロログプログラムを考えてみよう。もしこのプログラムの意味が sequencing を前提としないものならばこのプログラムは

$$\begin{array}{ll} P(x) & \\ P(f(x)) \Leftarrow Q(x) & \text{(II)} \end{array}$$

と等価でなければならない。プログラムⅡは sequencing のもとでは

$$P(x)$$

と等価であって明らかにプログラムIと異なる。
このようにプロログプログラムでは sequencing は本質的である。その際例えば

$$P(f(x)) \Leftarrow Q(x)$$

なる節が適用できないことを判別する必要がある。

$$\{P(f(x)) \mid x \in H\} \cap y = \emptyset ?$$

の判別である。答えが yes ならば z_1 を実行し no ならば z_2 を行なうわけである。このような関数は集合の包含関係に関して単調性を満たさないのでこういう関数を Scott の意味での連続関数と見なす場合には多少の工夫を必要とするであろう。

まず有限木の集合FTを次のように帰納的に定義する。

〔定義 1〕

FTは次の条件i) ii)を満たす最小集合

i) $C_n \in FT$　但し　$n \geq 1$

ii) $\alpha_1 ----, \alpha_m \in FT$ のとき

$$f_m^n(\alpha_1, ----, \alpha_m) \in FT$$
但し $1 \leq n \leq N,\quad 1 \leq m \leq N$

ここで C_n は定数記号 f_m^n は関数記号と呼ばれる。

FTは自然数全体と一対一に対応するのでFTを考えることと自然数全体 ω を考えることは同じであるが、FTを導入した方が話が簡単になるのである。

FTの巾集合にScott位相を入れたものをP(FT)と書くことにする。即ち

〔定義 2〕

α をFTの有限部分集合とするとき

$\text{above}(\alpha) = \{x \mid x \supset \alpha,\ x \subset FT\}$　と定義する。

$$\mathcal{I}_0 = \{\text{above}(\alpha) \mid x \supset \alpha,\ x \subset FT\}$$

を開集合の基とする位相をFTの巾集合P(FT)に入れるものとする。

例1.　$p = f_2^1,\quad f = f_1^1$　として

$$(p(x, f(x)) \Leftarrow p(f(x), x))(y)$$
$$= \{p(f(x), x) \mid p(x, f(x)) \in y\}$$

と定めると $(p(x, f(x)) \Leftarrow p(f(x), x))$ なる関数はP(FT)上連続

失敗の判定が連続となるようなモデルを構成するために$P(FT)$のレトラクトを考えよう。

〔定義3〕

有限木$F \in FT$が定数 $c_{n_1}, ----, c_{n_r}$, $n_1, ----, n_r > N$ を含むとき

$$F = F(c_{n_1}, ----, c_{n_r})$$

と書くことにする。

このとき

$$b : FT \to P(FT);$$
$$b(F(c_{n_1}, ----, c_{n_r})) = \{F(x_1, ----, x_r) \mid x_1, ----, x_r \in FT\}$$

と定める。

更に $\bar{b} : P(FT) \to P(FT);$

$$\bar{b}(x) = \bigcup_{F \in x} b(F)$$

〔命題 1〕

$\bar{b} : P(FT) \to P(FT)$は連続なレトラクション写像

指数N以下の定数のみからなる有限木の全体を$FT(N)$と書くことにする。

〔命題 2〕

$$r : P(FT(N)) \times \bar{b}(P(FT)) \to P(FT(N)) \times \bar{b}(P(FT))$$

$$; \quad r(\langle x, y \rangle) = \begin{cases} \langle x, y \rangle & x \cap y = \emptyset \\ \top & \text{else} \end{cases}$$

により定まる写像rはレトラクション写像である。

〔定義 4〕

空間Eを

$$E = r(P(FT(N)) \times \bar{b}(P(FT)))$$

と定める。

Eの要素を$\langle x, y \rangle$とするときxは正な情報をyは負な情報を担っているものと考える。

〔命題 3〕

$u \in P(FT(N))$ に対し $C(u)$ を

"$\langle u, C(u) \rangle$が $E \setminus \{\top\}$で極大"

なる条件を満たすものとするとき

$C(u)$ は写像

（証明）

$\langle u, v_1 \rangle$ ，$\langle u, v_2 \rangle$ が極大とすると

$\langle u, v_1 \cup v_2 \rangle$ も極大

故に

$$v_1 = v_1 \cup v_2 = v_2$$

Q.E.D.

例2.

$$\| p(x, f(x)) \Leftarrow p(f(x), x) \| (\langle u, v \rangle)$$

$$= \langle (p(x, f(x)) \Leftarrow p(f(x), x))(u), \\ (p(x, f(x)) \Leftarrow p(f(x), x))(v) \\ \cup C(\{ p(f(x), x) \mid x \in FT(n) \}) \rangle$$

と定めると $\| p(x, f(x)) \Leftarrow p(f(x), x) \|$ は連続

例3.

$$fail\ (p(f(x)), z)(\langle u, v \rangle)$$

$$= \begin{cases} z & v \supset \{ p(f(x)) \mid x \in FT \} \text{ のとき} \\ \emptyset & else \end{cases}$$

と定めると $fail\ (p(f(x)), z)$ は連続

例4.

$$\| g(x, f(x), y) \Leftarrow g(f(x), x, y) \| (\langle u, v \rangle)$$

$$= \langle (g(x, f(x), y) \Leftarrow g(f(x), x, y))(u),$$

$$\bigcup_{\substack{\alpha(c) \in FT^s \\ c = c_{n_1}, \cdots, c_{n_r} \\ n_1, \cdots, n_r > N}} fail\ (g(\alpha(x), f(\alpha(x)), y), \bigcup_{x \in FT^r} g(f(\alpha(x)), \alpha(x), y))(v)$$

$$\cup\ C(\{ g(f(x), x) \mid x \in FT(N) \}) \rangle$$

と定めると

$$\| g(x, f(x), y) \Leftarrow g(f(x), x, y) \|$$ は連続関数

例 2, 3, 4 に示したように Prolog の節は失敗も含めて E 上での連続関数と解釈できるであろう。本節では更に空間 E が reflexive であり、従って Prolog の semantics を E 上で規定できることも述べておく。

3. 空間 E_0

空間 E 上では Prolog の節の関数解釈は否定情報の取扱いのために簡明さを欠いていた。そこで空間 E の部分空間を撰んで、より簡明な関数解釈を与えたい。

〔定義 5〕

$$E_0 = \{x \mid x \text{ は } E \setminus \{T\} \text{ で極大}\}$$

とする。

〔定義 6〕

$P(FT(N))$ に

$$\mathcal{I}_1 = \{ \text{above}(\alpha) \cap \text{under}(\beta^c) \mid \alpha \text{ は } FT(N) \text{ の有限集合}, \ \beta \text{ は } {}^{\exists}\beta' : FT \text{ の有限集合}, \ \beta = \bar{b}(\beta') \cap FT(N) \}$$

で生成される位相を入れる。

〔命題 4〕

$$I : E_0 \longrightarrow P(FT(N)); \quad \langle x, y \rangle \longmapsto x$$

とすると I は位相同型

（証明）

I が一対一であることは明白

$$\langle x, y \rangle \in \text{above}(\alpha) \cap \text{above}(\beta)$$
$$\Leftrightarrow \quad x \in \text{above}(\alpha) \wedge y \supset \beta$$

$$y \supset \beta \Rightarrow y \cap FT(N) \supset \beta \cap FT(N)$$
$$\Rightarrow x \cap \beta \cap FT(N) = \emptyset$$
$$\Rightarrow x \subset (\beta \cap FT(N))^c$$

即ち $\langle x, y \rangle \in \text{above}(\alpha) \cap \text{above}(\beta)$

$$\Rightarrow x \in \text{above}(\alpha) \cap \text{under}((\beta \cap FT(N))^c)$$

逆に　$x \in \mathrm{above}(\alpha) \cap \mathrm{under}((\beta \cap FT(N))^c)$

とするとき

$$x \cap \beta \cap FT(N) = \emptyset$$

明らかに　$x \cap (\beta \setminus FT(N)) = \emptyset$

従って　$x \cap \beta = \emptyset$

$\langle x, y \rangle \in E_0$　ならば　y は

$$x \cap y = \emptyset$$

を満たすものの中で極大だから

$$y \supset \beta$$

即ち　$\langle x, y \rangle \in \mathrm{above}(\alpha) \cap \mathrm{above}(\beta)$

Q.E.D.

この命題より以下では $P(FT(N))$ と E_0 を同一視する。
E_0上では Prolog のインタプリター記述に於いて必要となる関数は次の例5，例6に見るように簡明な解釈を得，連続である。

例5

$z \in E_0$ のとき

$$(\mathrm{fail}(p(f(x)), z) \mid E_0)(u)$$

$$= \begin{cases} z & u^c \supset \{p(f(x)) \mid x \in FT(N)\} \text{ のとき} \\ \text{未定} & \text{else} \end{cases}$$

例6

$$(\| g(x, f(x), y) \Leftarrow g(f(x), x, y) \| \mid E_0)(u)$$

$$= (g(x, f(x), y) \Leftarrow g(f(x), x, y))(u)$$

最後に空間E_0についていくつかの性質を述べよう。

〔命題5〕

E_0の開集合で連続な関数は E 上の連続関数に拡張できる。

〔命題 6〕

E_0上で下から半連続な関数は粗な集合を除いて
E上の連続関数に拡張できる。

参考文献

[1] 大谷木重夫 「Prologの意味論」電子技術総合研究所彙報 第46巻第5,6号 昭和57年 5,6 月.(1982)

[2] 大谷木重夫 「非単調関数を含む自己適用可能な関数領域」電子技術総合研究所彙報 第47巻第2号 昭和58年 2月 (1983)

[3] Scott, D 「λ-Calculus and recursion theory」 prc. 3rd Scandinavian Logic Symposium pp 154-193 North Holland (1972)

[4] Scott, D 「Related theories of the λ-Calculus」 Essays on Combinatory logic lambda calculus and Formalism pp 402-448 North Holland (1980)

[5] Plotkin, G.D 「T^{ω} as a universal domain」 J. computer and system sciences vol 17 No2 pp 209-236 (1978)

[6] Apt, K.R. & van Emden 「Contributions to the theory of logic Programing」 Journal of A.C.M. vol 29 No3 pp 841-862 (1982)

[7] Kowalski, R. 「Predicate logic as a Programing language」 in proc. IFIP congress N.H. Stockholm (1974)

[8] Johnes, N.D Mycroft, A. 「Stepwise development of operational and denotational semantics for Prolog」

[9] Lassez, J.L. & Maher, M.J. 「Closures and Fairness in the semantics of programing logic」