数理解析研究所講究録
第 591 巻 1986 年 45-52

述語の微分について(その3)

西澤輝泰(新潟大学・経済)
(Teruyasu Nishizawa)

§1. はじめに

正規集合の微分を次々ととっていって、これを状態として有限オートマトンを構成する手法は、再帰プログラムを得るための最も古典的な自動プログラミング手法である、と考えられる。その本質は次のように把えられるであろう。即ち、あらかじめ場合分け $C_1, \cdots, C_n$ と変換 $f_1, \cdots, f_n$ を設定しておいて、現在の情報内容 x を判断し、もし場合 C_i に該当するならば変換 f_i を施す。このとき、場合分けと変換の組合せがある種の条件を満たしているならば、述語 P に対し、 $C_i(x) \longrightarrow (P(x) \equiv P_i(f_i(x)))$ となるような述語 $P_1, \cdots, P_n$ が定められる。各 P_i に対して同様に $P_{i1}, \cdots, P_{in}$ を定め、これをくり返す。この過程でもはや新しい述語が生成されなくなれば、ここに有限個の述語の再帰的な定義式が自然に得られることになる。[1]では、このような考え方によって純LISPプログラムの自動合成シス

テムを与えている。場合分けと変換はこれを対にして考えれば部分関数の適用であり，［2］，［3］ではこうした部分関数による述語の微分について述べている。［4］では更に，部分関数ベクトルによる述語の微分を定式化しているが，それによって得られる再帰プログラムは木オートマンに対応する。［2］，［3］，［4］における“微分”の定式化で得られる再帰プログラムは，計算過程で情報の書き込みができない。それを可能とするような拡張が本稿の目的である。その拡張された定式化は［4］の定式化に基礎をおいているので，次節で［4］の概要を述べる。ただし，定式化に若干の修正を加える。

§2．有限微分閉包

集合 W 上の1引数述語 P が，$W \to W^m$ の部分関数 $\vec{f}$ により微分可能とは，

$$(\forall x, y \in D(\vec{f}))\,[\,\vec{f}(x)=\vec{f}(y) \longrightarrow P(x) \equiv P(y)\,]$$

が成り立つことをいう。ここに $D(\vec{f})$ は $\vec{f}$ の定義域である。m を $d(\vec{f})$ で表す。$m=0$ も許容することとし，W^0 は仮空の要素 ε 唯一つからなる集合 $\{\varepsilon\}$ であるとする。これにより，$d(\vec{f})=0$ なる $\vec{f}$ で微分可能とは，

$(\forall x, y \in D(\vec{f}))\quad P(x) \equiv P(y)$

が成り立つことである。

$d(\vec{f}) = m$ なる $\vec{f}$ で述語 P が微分可能であるとき，その微分 $\partial_{\vec{f}} P$ は $\overline{W}$ 上の m 引数述語として次により定められる。

$$(\partial_{\vec{f}} P)(x_1, \cdots, x_m)$$
$$\longleftrightarrow (\exists y \in D(\vec{f}))\,[\,\vec{f}(y) = (x_1, \cdots, x_m) \wedge P(y)\,]$$

$d(\vec{f}) = 0$ のとき，$\partial_{\vec{f}} P$ は定数（true 又は false）である。

$\overline{W} \to \overline{W}^m$ $(m = 0, 1, 2, \cdots)$ の部分関数の有限集合 $\mathcal{F}$ で，$\bigcup_{\vec{f} \in \mathcal{F}} D(\vec{f}) = \overline{W}$ なるようなFすべてからなる族を $\mathcal{C}_{\overline{W}}$ で表すことにする。$\mathcal{F} \in \mathcal{C}_{\overline{W}}$ について，述語 P が任意の $\vec{f} \in \mathcal{F}$ により微分可能ならば，次式が成り立つ。

$$P(x) \longleftrightarrow \bigvee_{\vec{f} \in \mathcal{F}} (x \in D(\vec{f}) \wedge (\partial_{\vec{f}} P)(\vec{f}(x)))$$

$\overline{W}$ 上の m 引数述語 $\mathbb{P}$ が，$\overline{W}$ 上の 1 引数述語からなる族 Φ を台とする変数分離形であるとは，有限個の添字 k（$k = 1, \cdots, l$ とする）と各 $i = 1, \cdots, m$ に対し，Φ の元であるか又は定数 true であるような $Q_i^{(k)}$ が存在して，$\mathbb{P}(x_1, \cdots, x_m) \equiv \bigvee_{k=1}^{l} \bigwedge_{i=1}^{m} Q_i^{(k)}(x_i)$

が成り立つことをいう。

W 上の1引数述語 $P_0, \dots, P_n$ と C_W の元 $\mathcal{F}_0, \dots, \mathcal{F}_n$ からなる系 $\{P_0, \dots, P_n ; \mathcal{F}_0, \dots, \mathcal{F}_n\}$ が有限微分閉包をなすとは，各 P_i が任意の $\vec{f} \in \mathcal{F}_i$ により微分可能で，$\partial_{\vec{f}} P_i$ が $\{P_0, \dots, P_n\}$ を台とする変数分離形となっていることをいう。このとき，

$$
\begin{array}{l} i=0 \\ \\ \\ n \end{array}
\left\{
\begin{array}{l}
P_i(x) \longleftrightarrow \displaystyle\bigvee_{\vec{f} \in \mathcal{F}_i} \Big(x \in D(\vec{f}) \wedge \bigvee_{k=1}^{l_{i,\vec{f}}} \bigwedge_{j=1}^{d(\vec{f})} Q_{ij}^{(k)}(f_j(x)) \Big) \\
\\
\\
\end{array}
\right.
$$

となる。

（ただし，$(\partial_{\vec{f}} P_i)(x_1, \dots, x_m) \equiv \bigvee_{k=1}^{l_{i,\vec{f}}} \bigwedge_{j=1}^{d(\vec{f})} Q_{ij}^{(k)}(f_j(x))$ であり，$Q_{ij}^{(k)}$ は $P_0, \dots, P_n$ のいずれかか又は true である。また，$d(\vec{f}) = 0$ のときはこの部分は定数 $\partial_{\vec{f}} P_i$ である。）

これは，$P_0, \dots, P_n$ を計算する非決定性再帰プログラムである。

W が半順序 $<$ に関して整礎集合であり，各 $\vec{f} = (f_1, \dots, f_m) \in \mathcal{F}_i$ が $f_j(x) \lneq x$ なる条件（これを下降条件という）を満たせば，このプログラムは必ず停止する。これは，root-to-frontier の非決定性木オートマトンに対応し，このときの計算過程を表す木（節点の名札は各 $\mathcal{F}_i$ の元。名札 $\vec{f}$ をもつ節点 α は $m = d(\vec{f})$ 個の子を持ち，節点 α での

情報内容（Wの元）がxであるとき，子節点の名札は，

$f_{j}(x) \in D(\vec{f}_{ij})$なる$\vec{f}_{i1}, \ldots, \vec{f}_{im}$である。）の集合の族は，木の正規集合の族に一致する。

もし，$\mathcal{F}_{i}$に属するすべての$\vec{f}$について，$D(\vec{f})$が互いに素であり，また各$l_{i,\vec{f}}$が1であれば，これは決定性の再帰プログラムであり，決定性の木オートマトンに対応する。

§3. 有限記述的無限微分閉包

ここで述べる定式化は，記述の複雑性を避けるため，ある種の決定性を受け入れた形で記述する。前節と同様な非決定性を許容する形に一般化することは容易である。

可算無限集合Γと，Γの有限分割$\Pi = \{\Gamma_{\delta} ; \delta \in \Delta\}$を与える。（前節における系$\{P_{\lambda}, \mathcal{F}_{\lambda} ; \lambda \in \Lambda\}$（ただし，$\Lambda$は添字の有限集合，$P_{\lambda}$は$W$上の$1$引数述語，$\mathcal{F}_{\lambda}$は$\mathcal{C}_{W}$の元）は，ここでは無限個の述語を含む系$\{P_{\lambda,\gamma}, \mathcal{F}_{\lambda} ; \lambda \in \Lambda, \gamma \in \Gamma\}$に拡張される。$\Gamma$の有限分割$\Pi$は有限的記述を可能とするため導入された。）

Λを添字の有限集合とし，$\{\mathcal{F}_{\lambda} ; \lambda \in \Lambda\} \subset \mathcal{C}_{W}$を与える。各$\lambda \in \Lambda$, $\delta \in \Delta$と各$\vec{f} \in \mathcal{F}_{\lambda}$に対し，$d(\vec{f}) = m$として，$\Gamma \rightarrow \Gamma^{m}$の関数$\varphi_{\lambda,\delta,\vec{f}}$を対応させる。

系 $\mathfrak{A}$：$\{P_{\lambda,\gamma},\ \mathfrak{F}_\lambda\ ;\ \lambda\in\Lambda,\ \gamma\in\Gamma\}$ & $(\Gamma, \Pi, \vec{\varphi})$ （ただし、各 $P_{\lambda,\gamma}$ は $\mathbb{W}$ 上の 1 引数述語）が有限記述的無限微分閉包（単に、微分閉包と略称する）を構成する、ということを次のように定める。即ち、

各 $P_{\lambda,\gamma}$ は各 $\vec{f}\in\mathfrak{F}_\lambda$ により微分可能で、$d(\vec{f})=m$, $\gamma\in\Gamma_\delta$, $\vec{\varphi}_{\lambda,\delta,\vec{f}} = (\psi_1,\cdots,\psi_m)$ とすると、$\lambda, \delta, \vec{f}$ により $\lambda_1,\cdots,\lambda_m\in\Lambda$ が定まって、

$$(\partial_{\vec{f}} P_{\lambda,\gamma})(x_1,\cdots,x_m) \equiv \bigwedge_{i=1}^{m} P_{\lambda_i,\psi_i(\gamma)}(x_i)$$

となる。（このような λ_i を $\xi(\lambda,\delta,\vec{f},i)$ で表すことにする。）

これにより、系 $\mathfrak{A}$ が微分閉包をなすときは、次のような形の再帰プログラムが得られる。（ただしプログラムとしては $P_{\lambda,\gamma}(x)$ は $P_\lambda(x,\gamma)$ として取扱わなければならない。）

$$\begin{array}{l} \lambda\in\Lambda \\ \gamma\in\Gamma \end{array}\left\{ P_{\lambda,\gamma}(x) \longleftrightarrow \bigvee_{\vec{f}\in\mathfrak{F}_\lambda} \left(x\in D(\vec{f}) \wedge \bigwedge_{i=1}^{d(\vec{f})} P_{\lambda_i,\psi_i(\gamma)}(f_i(x))\right)\right.$$

（ただし、$d(\vec{f})=m$ として、$\vec{f}=(f_1,\cdots,f_m)$, $\vec{\varphi}_{\lambda,\delta,\vec{f}} = (\psi_1,\cdots,\psi_m)$, $\lambda_i=\xi(\lambda,\delta,\vec{f},i)$ である。）

前節のように $\vec{f}$ の下降条件を導入すれば、このプログラムは必ず停止する。また，$\mathfrak{F}_\lambda$ のすべての $\vec{f}$ について $D(\vec{f})$ が互いに素なら，決定性の再帰プログラムである。

このプログラムの前提となる計算は，各 $\vec{f}$ 及び 各 $\mathfrak{P}_{\lambda,\delta,\vec{f}}$ の計算と，$\gamma \in \Gamma_\delta$ なる δ の決定手続である。

前節のような非決定性を許容する形で自然に拡張すれば，このようなプログラムは西澤の general indexed grammar に対応する。従って例えば，Γ を pushdown stack の内容の集合（即ち語の集合），Π を top symbol による Γ の分割，$\mathfrak{P}_{\lambda,\delta,\vec{f}}$ を，pushdown stack の標準操作に限定すれば，こうしたプログラムの計算過程を表す木の集合の族は，Aho の indexed grammar の生成する木の集合の族に一致することになる。

§. 引用文献

[1] M. Nagai & T. Nishizawa, A system for automatically synthesizing pure LISP programs, Proc. 6-th IBM symp. on MFCS, 1981

[2] 西澤輝泰，整礎集合上の述語の微分について，1981年夏の LA シンポジウム

[3] T. Nishizawa, Automaton Programs and Regular Functional Expressions — On An Extension of Derivatives, Bull. Informatics and Cybernetics, vol.21, No.1~2, Research Assoc. of Statistical Sci., 1984
[4] 西澤輝泰，述語の微分について（その2），1985年夏のLAシンポジウム