例からの関数合成における関数の性質

2C-4

犬塚 信博　　石井 直宏
名古屋工業大学

1. はじめに　これまで、リスト(S式)からリストへのＬＩＳＰの関数を、有限個の入出力例から帰納的に構成する方法について、Summers[1]の方法 をもとに考察してきた。本報告では、この方法に与えるべき例の性質の面から検討を加える。そのため、関数を導くのに適当な例を典型例として定義し、意図する関数を導出するために、そうした例を与える必要があることを示す。

2. 導出方法　一般に次の手順により、入出力例の集合 $\{x_i \to y_i \mid 1 \leqq i \leqq n\}$ から、それが示すような振舞いをする関数を導く。「→」は、入力と出力の区切りである。

(1) $p_i[x_j] = \begin{cases} \mathrm{true}(i=j) \\ \mathrm{false}(i \neq j) \end{cases}$ なる述語 $p_1, \cdots, p_n$ を求める。

(2) $ff_i[x_i] = y_i$ なる関数fragment $ff_1, \cdots, ff_n$ を求める。
これらはcar, cdr, consの１変数合成関数である。

(3) $p_1, \cdots, p_n$ に繰り返し的な規則性を見出だす。

(4) $ff_1, \cdots, ff_n$ に繰り返し的な規則性を見出だす。

(5) (3), (4)の規則性が $i \geqq n$ でも成り立つとして、この関数をプログラムとして生成する。

3. 典型例　本報告ではＳ式の集合Ｓ上の関数を考えているが、議論の簡単化のため S_0 に限定する。つまり、S_0 は点対を用いなくても表現できるＳ式である。また、S_0 の上の置換を $\sigma = \{a_1/s_1, \cdots, a_n/s_n\}$ と表わし、$s\sigma$ により、アトム $a_1, \cdots, a_n$ をＳ式 $s_1, \cdots, s_n$ に置き換えることを意味する。

リストを反転する関数では、例(a b)→(b a)と((a b) (c d))→((c d) (a b))は同じことを意味しており、このようなものを同じ型であると称してまとめることにする。

【定義１】関数ｆに対して定まる同値関係 $=_f$ による S_0 の同値類をそれぞれ入力の型と呼ぶ。$s \in S_0$ の属する入力の型（つまり、同値類）を $[s]_f$ と書く。但し、$=_f$ は次の関係 $\simeq_f$ の推移・対称閉包である。

$s_1 \simeq_f s_2 \Leftrightarrow \exists \sigma,\ s_1\sigma = s_2 \wedge f[s_1]\sigma = f[s_2]$ □

ｓと s' が同じ型に属している場合、これらが与える入出力関係は同じことを表わしていると考えられる。

ある型の中で最も簡単なものとして典型的な入力を考えるため、その尺度として半順序≦を導入する。≦は擬順序≦に付随し、これ付随する同値関係Ｒ（$s_1 R s_2 \Leftrightarrow s_1 \leqq s_2 \wedge s_2 \leqq s_1$）の同値類上の半順序である。また、form: $s \mapsto [s]_R$ とする。但し、ωは任意のアトム、ａ, ｂ, ｃ, ｄは任意のｓ式とすると、≦は、

$\omega \leqq a,\ a \leqq b \wedge c \leqq d \Rightarrow (a.c) \leqq (b.d)$。

これを用いて典型的入力を次のように定義する。

【定義２】関数ｆの各入力の型 $[s]_f$ について、form$[[s]_f]$ に、≦に関する最小元 $\tilde{s}$ が存在するとき、これをｆの典型的入力、そして、$\tilde{s} \to f[\tilde{s}]$ を典型例という。また、ｆの典型的な入力全体の集合を T_f で表す。□

4. 典型的な例の集合　関数ｆを導出するために必要な例の集合典型性として以下の条件を与える。

【定義３】例の集合 $\{x_i \to y_i \mid 1 \leqq i \leqq n\}$ が典型的とは、以下の条件を満たすとき言う。但し $IN = \{x_i \mid 1 \leqq i \leqq n\}$。

(1) form[IN]が≦において全順序集合を成す。

(2) $IN - T_f = \phi$。

(3) $\forall t_1 \in IN,\ \forall t_2 \in T_f - IN,\ \mathrm{form}[t_2] \not\leqq \mathrm{form}[t_1]$

これに対し、次のような関数ｆの典型性条件を考える。

【条件】次の(1)～(3)を、関数ｆの典型性条件と呼ぶ。

(1) ｆの各入力の型のformによる像は、それぞれ≦に関して最小元を持つ。

(2) T_f のformによる像は、≦に関して全順序を成す。

(3) ｆの任意の入力の型α, βに対して、
$\forall s_1 \in \alpha,\ \forall s_2 \in \beta,\ \mathrm{form}[s_1] \not\leqq \mathrm{form}[s_2]$
$\forall s_1 \in \alpha,\ \forall s_2 \in \beta,\ \mathrm{form}[s_2] \not\leqq \mathrm{form}[s_1]$。□

(1)(2)は、任意の大きさの典型的集合の存在を保証するものであり、(3)は次の関数fragmentと述語の性質等により、導出方法の部分正当性を保証するものである。

【性質１】関数ｆに対し $\forall s \in S$, $\forall ff$ (関数fragment), $ff[s] = f[s] \Rightarrow \forall \hat{s} \in [s]_f,\ ff[\hat{s}] = f[\hat{s}]$。□

【性質２】$s_1, s_2 \in T_f$ のとき、$p[s_1] = T,\ p[s_2] = \perp \Rightarrow \forall \hat{s}_1 \in [s_1], \forall \hat{s}_2 \in [s_2],\ p[\hat{s}_1] = T,\ p[\hat{s}_2] = \perp$ □

5. おわりに　ＬＩＳＰプログラムの導出において、例の持つべき性質、そうした例を持つ関数の満たすべき条件について検討した。このことは、この種の方法により導出できる関数を求めるために有用であると考えている。

(1)P. D. Summers:"A Methdology for LISP Construction from examples", JACM 24, pp. 161-175, 1977.

Properties of Functions in Function Synthesis from Examples
Nobuhiro INUZUKA　Naohiro ISHII
Nagoya Institute of technology