論理プログラムと計算可能性

専門共通科目「計算モデル論」における論理プロクラムの計算可能性のまとめ

山田 敬三

1. はじめに

本稿では，本学部で 2 年生前期に開講される専門共通科目の一つである「計算モデル論」の内容について，その一部をまとめる．この講義は，大きく 2 つのパートに分かれており，前半は，チューリング機械を中心とした機械モデルについて講義し，後半は，論理モデルについてホーン節を用いたプログラミング言語 Prolog を例に取り上げながら講義する．

本稿では，ホーン節に基づくプログラミング言語について，帰納的関数と同じ計算をするホーン節プログラムの構成法を確認し，これに基づく四則計算をはじめとする，いくつかの計算を，ホーン節プログラムによって表す．まず，はじめに，足し算，引き算，掛け算，剰余計算の構成法を確認し，その過程で，与えられた 2 つの数が同じか否かを調べる相当関数，および与えられた数が 0 か否かを調べる符号関数を計算するホーン節プログラムを構成する．次に，原始帰納的関数として割り算の商を求める関数を構成に基づいたホーン節プログラムを示し，最後に，平方根を求める帰納的関数の構成に基づいたホーン節プログラムを示す．

2. 原始帰納的関数と帰納的関数

本章では，原始帰納的関数と，帰納的関数について確認する．

定義 1 帰納的関数

以下では，m,nは 1 以上の自然数とする．

1. 定数関数：$Z(x)=0$は，帰納的関数である．
2. 後者関数：$S(x)=(x\text{の後者})$は，帰納的関数である．
3. 射影関数：$U_i^n(\overrightarrow{x_n})=x_i\ (1\le i\le n)$は帰納的関数である．
4. 合成：$h_i:\mathbb{N}^n\to\mathbb{N}\ (1\le i\le m)$および$g:\mathbb{N}^m\to\mathbb{N}$が，それぞれ帰納的関数のとき，
$$f(\overrightarrow{x_n})=g\left(\overrightarrow{h_m(\overrightarrow{x_n})}\right)$$
は帰納的関数である．
5. 原始帰納：$g:\mathbb{N}^{n-1}\to\mathbb{N}$, $h:\mathbb{N}^{n+1}\to\mathbb{N}$がともに帰納的関数であるとき，
$$\begin{cases} f(\overrightarrow{x_{n-1}},0)=g(\overrightarrow{x_{n-1}}) \\ f\left(\overrightarrow{x_{n-1}},S(y)\right)=h\left(\overrightarrow{x_{n-1}},y,f(\overrightarrow{x_{n-1}},y)\right)\end{cases}$$
は帰納的関数である．なお，$n=1$のとき，gは定数を表す．
6. 最小化：$g(\overrightarrow{x_n},y):\mathbb{N}^{n+1}\to\mathbb{N}$が帰納的関数のとき，$g$から最小化によって構成される関数
$$f(\overrightarrow{x_n})=\mu y\left(g(\overrightarrow{x_n},y)\right)$$
すなわち，
$$f(\overrightarrow{x_n})=\begin{cases}\min\{y|g(\overrightarrow{x_n},y)=0\}; \\ g(\overrightarrow{x_n},y)=0\text{ となる}y\text{が存在する} \\ \text{未定義;そうでないとき}\end{cases}$$
は帰納的関数である．
7. 上の 1.～6. で構成される関数だけが帰納的関数である．

また，帰納的関数のうち，6. の最小化を使わずに構成される関数を原始帰納的関数という．

3. 論理プログラムの計算可能性

3.1. 健全性と完全性

ある形式的体系における定理がすべて真であるとき，その形式的体系を健全であるという．また，すべての真である論理式が証明可能である（定理となりうる）とき，その形式的体系は完全であるという．

3.2. 論理プログラムの計算可能性

まず，自然数kを，後者関数を用いて$s^k(0)$と表す．すなわち，$s^k(0)=\underbrace{s(s(\cdots \mathrm{s}}_{k\text{個}}(0)\cdots))$であり，$k=0$のときは0とする．

定義 2 述語表現可能

関数fを自然数上のn引数関数，Γを論理プログラ

ム，p_fを$n+1$引数述語とする．このとき，次の 1.，2. が同値ならば，fはΓとp_fで述語表現可能という．

1. $f(x_1, x_2, \ldots, x_n) = k$
2. Γのもとで$p_f(s^{x_1}(0), s^{x_2}(0), \ldots, s^{x_n}(0), y)$を目標節とする線形反駁の導出列が存在し，そのときの単一化代入において，yは$s^k(0)$に置き換えられる．

このとき，次の定理が成り立つ．

定理 1

帰納的関数は，述語表現可能である．

（証明）定数関数，後者関数，射影関数は，それぞれ次のホーン節をΓに加えることで述語表現可能である．

定数関数：$p_z(x, 0) \leftarrow$

後者関数：$p_s(x, s(x)) \leftarrow$

射影関数：$p_{ui}(x_1, \ldots, x_i, \ldots x_n, x_i) \leftarrow$

合成により構成される関数fについて，n引数関数$h_1, h_2, \ldots h_m$が$n+1$引数述語$p_{h1}, p_{h2}, \ldots, p_{hm}$で，それぞれ述語表現可能であり，かつ$m$引数関数$g$が$m+1$引数述語$p_g$で述語表現可能であるとき，次の$p_f$によって$f$は述語表現可能である：

$$p_f(\overrightarrow{x_n}, z) \leftarrow \overrightarrow{p_{hm}(\overrightarrow{x_n}, y_m)}, p_g(\overrightarrow{y_m}, z)$$

原始帰納法により構成される関数fについて，n引数関数gと$n+1$引数関数hが，それぞれ$n+1$引数述語p_gと$n+2$引数述語p_hで述語表現可能であれば，次のp_fによってfは述語表現可能である：

$p_f(\overrightarrow{x_{n-1}}, 0, z) \leftarrow p_g(\overrightarrow{x_{n-1}}, z)$

$p_f(\overrightarrow{x_{n-1}}, s(y), z) \leftarrow p_f(\overrightarrow{x_{n-1}}, y, w), p_h(\overrightarrow{x_{n-1}}, y, w, z)$

最小化により構成される関数fについて，$n+1$引数関数gが$n+2$引数述語p_gで述語表現可能であれば，次のp_fによってfは述語表現可能である：

$p_f(\overrightarrow{x_n}, z) \leftarrow p_g(\overrightarrow{x_n}, 0, w), p_q(\overrightarrow{x_n}, 0, w, z)$

$p_q(\overrightarrow{x_n}, y, 0, y) \leftarrow$

$p_q(\overrightarrow{x_n}, y, s(v), z) \leftarrow p_g(\overrightarrow{x_n}, s(y), w), p_q(\overrightarrow{x_n}, s(y), w, z)$

■

4. 四則演算を行う述語の構成

本章では，簡単な計算をする，いくつかの述語を前章の定理に沿って構成する．

4.1. 足し算を行う述語

加算関数addは，原始帰納法を用いて，

$$\begin{cases} add(x, 0) = x \\ add(x, s(y)) = s(add(x, y)) \end{cases}$$

と定義できるので，ホーン節を用いて表すと，

$add(x, 0, x) \leftarrow$

$add(x, s(y), z) \leftarrow add(x, y, w), p_s(w, z)$

となる．以降，$add(x, y)$を$x + y$と表す．

4.2. 前者を求める述語

前者関数Pは，原始帰納法を用いて，

$$\begin{cases} P(0) = 0 \\ P(s(x)) = x \end{cases}$$

と定義できるので，ホーン節を用いて表すと，

$P(0,0) \leftarrow$

$P(s(x), x) \leftarrow$

となる．

4.3. 引き算を行う述語

減算関数$minus$は，原始帰納法を用いて，

$$\begin{cases} minus(x, 0) = x \\ minus(x, s(y)) = P(minus(x, y)) \end{cases}$$

と定義できるので，ホーン節を用いて表すと，

$minus(x, 0, x) \leftarrow$

$minus(x, s(y), z) \leftarrow minus(x, y, w), P(w, z)$

となる．以降，$minus(x, y)$を$x \ominus y$と表す．

4.4. 掛け算を行う述語

乗算関数$multi$は，原始帰納法を用いて，

$$\begin{cases} multi(x, 0) = 0 \\ multi(x, s(y)) = add(multi(x, y), x) \end{cases}$$

と定義できるので，ホーン節を用いて表すと，

$multi(x, 0, x) \leftarrow$

$multi(x, s(y), z) \leftarrow multi(x, y, w), add(w, x, z)$

となる．以降，$multi(x, y)$を$x \times y$と表す．

4.5. 正か 0 かを判定する述語

正か 0 かを判定する関数$sign$は，つぎのような関数である．

$$sign(x) = \begin{cases} 0; x = 0 \\ 1; x \geq 1 \end{cases}$$

この関数は原始帰納的関数であり，つぎのように定義される：

$$sign(x) = 1 \ominus (1 \ominus x)$$

この関数はホーン節を用いて表すと，

$sign(x, y) \leftarrow minus(s(0), x, w), minus(s(0), w, y)$

となる.

4.6. 等価性を判定する述語

等価性判定関数$equal$は，つぎの関数である：

$$equal(x,y)=\begin{cases}0;\ x=y\\1;\ x\neq y\end{cases}$$

この関数は原始帰納的関数であり,

$$equal(x,y)=sign\big((x\ominus y)+(y\ominus x)\big)$$

と定義される．この関数は，ホーン節を用いて表すと，つぎのようになる：

$$equal(x,y,z)\leftarrow minus(x,y,w_1),minus(y,x,w_2),$$
$$add(w_1,w_2,w_3),sign(w_3,z)$$

また，$equal$とは逆に，xとyが異なるときに 0 を返し, 同じときに 1 返す関数$\overline{equal}$も原始帰納関数であり,

$$\overline{equal}(x,y)=1\ominus equal(x,y)$$

と定義される．この関数は，ホーン節を用いて表すと，つぎのようになる.

$$\overline{equal}(x,y,z)\leftarrow equal(x,y,w),minus(s(0),w,z)$$

4.7. 剰余を求める述語

剰余演算を行う関数modとは, つぎの関数である:

$$mod(x,y)=\begin{cases}x;y=0\\x\text{を}y\text{で割った余り};y\neq 0\end{cases}$$

この関数は原始帰納的関数であり,

$$\begin{cases}mod(0,y)=0\\w=s\big(mod(x,y)\big)\text{とすると}\\mod(s(x),y)=w\times equal(y,w)\end{cases}$$

と定義される．この関数は，ホーン節を用いて表すと,

$$mod(0,y,0)\leftarrow$$
$$mod(s(x),y,z)\leftarrow mod(x,y,w_1),p_s(w_1,w_2),$$
$$equal(y,w_2,w_3),multi(w_2,w_3,z)$$

となる.

4.8. 割り算を行う述語

除算を行う関数divは原始帰納的関数であり,

$$\begin{cases}div(0,y)=0\\div(s(x),y)=div(x,y)+\overline{equal}\Big(y,s\big(mod(x,y)\big)\Big)\end{cases}$$

と定義される. この関数はホーン節を用いて表すと,

$$div(0,y,0)\leftarrow$$
$$div(s(x),y,z)\leftarrow div(x,y,w_1),mod(x,y,w_2),$$
$$p_s(w_2,w_3),\overline{equal}(y,w_3,w_4),$$
$$add(w_1,w_4,z)$$

となる.

4.9. 平方根を求める述語

平方根を求める関数$root$は，つぎのように最小化を用いて定義される帰納的関数である：

$$root(x)=\mu y\big(equal(x,y\times y)\big)$$

この関数はホーン節を用いて表すと,

$$root(x,z)\leftarrow p_g(x,0,w),p_q(x,0,w,z)$$
$$p_g(x,y,z)\leftarrow multi(y,y,w),equal(x,w,z)$$
$$p_q(x,y,0,y)\leftarrow$$
$$p_q(x,y,s(v),z)\leftarrow p_g(x,s(y),w),p_q(x,s(y),w,z)$$

となる.

5. おわりに

本稿では，論理プログラムの計算可能性について述べ，帰納的関数は述語表現可能であることを確認した．また，いくつかの計算について，証明に沿って述語を構成した.

参考文献

[1] 猪股俊光, 山田敬三, “計算モデルとプログラミング”, 森北出版, 2019.

[2] 足立暁生, “計算機科学の基礎”, オーム社, 1987.