情報技術史論
（担当　栗原）

計算機の機構と計算理論

- 計算の考え方
- 帰納関数

参考文献
計算モデル論入門, 井田哲雄・浜名誠 著, サイエンス社

■ 計算の考え方： 計算とは何か？

「計算」と思われるものの例

(1) 固定ビット長で表された2つの2進数の和を求める

(2) 与えられた0と1からなる任意長の文字列が01の繰り返しであるか否かを判定する.

正規表現
$(01)^*$

(3) 任意の自然数 n を与えて，1から n までの和を求める.

(4) n が258以下で 2^n-1 が素数である数をすべて見いだす.

たとえば
$P = 9x^2-24y^5+z^2$

(5) 整数係数の多項式 P を与えて，方程式 P=0 が整数解をもつかどうかを判定する.（ヒルベルトの第10問題）

計算を機械で行えるのか？

早さや効率を考慮せず
計算可能性のみを考慮

機械によって「計算能力」が異なる

(1) 固定ビット長で表された2つの2進数の和を求める
有限の論理値の組合せ（真理値表）で十分
→ 組合せ回路

(2) 与えられた0と1からなる任意長の文字列が01の繰り返しであるか否かを判定する.
計算状態を表す有限のメモリが必要
→ 有限オートマトン（順序回路）

(3) 任意の自然数 n を与えて，1から n までの和を求める.
無限のスタックメモリが必要
→ プッシュダウン・オートマトン

(4) n が258以下で 2^n-1 が素数である数をすべて見いだす.
無限のテープ（またはRAM）が必要
→ チューリング機械
現代のコンピュータ

(5) 整数係数の多項式 P を与えて，方程式 P=0 が整数解をもつかどうかを判定する.（ヒルベルトの第10問題）
計算不能
→ これを計算できる機械はまだ発見・発明されていない

計算能力：小 → 大

計算可能性の概念(チャーチの提唱)

計算に関する3つの重要な概念(1930年代)

チューリング機械	ラムダ計算	帰納関数
(Turing machine)	(lambda calculus)	(inductive function)
チューリング	チャーチ, ロッサー	エルブラン, ゲーデル クリーネ
第3章	第7章	第5章

これらの計算機構は, 同じ計算能力をもつ
(互いに他をシミュレートできる)

現代のコンピュータ

チャーチの提唱

計算可能である＝帰納関数で定義できると考えよう.

この「提唱」は, 証明されたわけではない

■ 帰納関数

1 帰納関数の考え方
2 初期関数
3 合成
4 原始帰納
5 μ 演算子
6 最小化
まとめ

1　帰納関数の考え方

帰納関数

- 初期関数

という非常に基本的な関数から

- 合成
- 原始帰納
- 最小化

を用いて組み立てられた関数のこと.

自然数の集合Nを定義域と値域とするものとする.

自然数には 0 を含めることとする

2　初期関数

初期関数

- $succ(x) = x+1$　（後者関数）
- $zero(x) = 0$　（定数ゼロ関数）
- $u_i^n(x_1,\cdots,x_n) = x_i \quad 1 \leqq i \leqq n$　（射影関数）

【演習問題5.1】

恒等関数は初期関数である:

$$u_1^1(x) = x$$

3　合成

合成

任意の関数 $g, h_1, \cdots, h_m$ を以下のように組み合わせて，次の関数 f を定義する操作を合成と呼ぶ.

$$f(x_1,\cdots,x_n) = g(h_1(x_1,\cdots,x_n), \cdots, h_m(x_1,\cdots,x_n))$$

特に，n=m=1 のときは:

$$f(x) = g(h(x))$$

4　原始帰納

原始帰納

任意の関数 g, h を以下のように組み合わせて，次の関数 f を定義する操作を原始帰納と呼ぶ.

$$f(0, x_1,\cdots,x_n) = g(x_1,\cdots,x_n)$$

$$f(k+1, x_1,\cdots,x_n) = h(f(k, x_1,\cdots,x_n), k, x_1,\cdots,x_n)$$

特に，n=1のときは:

$$f(0, x) = g(x)$$

$$f(k+1, x) = h(f(k,x), k, x)$$

【例5.1】

$$\mathrm{plus}(0, x) = x$$

$$\mathrm{plus}(k+1, x) = \mathrm{succ}(\mathrm{plus}(k, x))$$

原始帰納関数

初期関数から合成と原始帰納の有限回の繰り返しによって得られる関数を原始帰納関数という.

５　μ演算子

μ演算子

任意の関数 g に対して，

$$\mu_m[g(x_1,\cdots,x_n,m) = 0]$$

の値を，$g(x_1,\cdots,x_n,m) = 0$ とする最小の自然数 m と定義する．

そのようなmが存在しなければ
gの値は未定義
（部分関数）

$g(x_1,\cdots,x_n,\ i)$ を $i=0,1,\cdots$ で計算していき，
i がmになって初めて，$g(x_1,\cdots,x_n,\ i) = 0$ となる．

そのようなmが存在しなければ
計算は停止しない
（部分的に計算可能）

Whileループで実現できる

6　最小化

最小化

任意の関数 g から，次の関数 f を定義する操作を**最小化**と呼ぶ.

$$f(x_1,\cdots,x_n) = \mu_m[g(x_1,\cdots,x_n,m) = 0]$$

【例5.3】

$$f(x) = \mu_m[\,2-x-m = 0\,]$$

$$f(0) = \mu_m[\,2-0-m = 0\,] = 2$$

$$f(1) = \mu_m[\,2-1-m = 0\,] = 1$$

$$f(2) = \mu_m[\,2-2-m = 0\,] = 0$$

$$f(3) = \mu_m[\,2-3-m = 0\,] = \perp$$（未定義）

まとめ　帰納関数

未定義の値を許す
（部分関数）

帰納関数

初期関数から合成，原始帰納，最小化の有限回の繰り返しによって得られる関数を帰納関数という.

帰納関数はチューリング機械により（部分的に）計算可能である.

逆に，チャーチの提唱により，
帰納関数として定義できない関数は
（どんな計算機構を用いても）計算不能である.

最小化を用いずに定義された帰納関数には未定義の値が存在せず，チューリング機械により（完全に）計算可能である.