計算できない問題について

Sakaé Fuchino (渕野 昌)

fuchino@isc.chubu.ac.jp

http://math.cs.kitami-it.ac.jp/~fuchino/

2005年7月13日（水） 中部大学 総合科目 情報工学科一年のクラスのための補講

$\mathbb{N}$ で自然数の全体を表します．計算機科学では，自然数は 0 から始まっていることにすることが多いので，ここでもそれに従うことにします．つまり，

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

“自然数” は英語で “natural numbers” というので，この頭文字の N をとって表すことにしているわけです．

f が $\mathbb{N}$ から $\mathbb{N}$ への 関数(記法: $f : \mathbb{N} \to \mathbb{N}$) とは，$\mathbb{N}$ の各要素 n に，対応する $\mathbb{N}$ の要素 $f(n)$ を与える与え方の指定となっていること

“f” は単に，ここで，この場かぎりの使いすてで使っている名前で f でなくて g でも h でも ϕ でも χ でも何でもいいのですが，これも “関数” が英語では “function” なので，とりあえず，この頭文字を使ってみているだけです．

$\mathbb{N}$ から $\mathbb{N}$ への関数の例

例 1. f を自然数 n に対して，n 番目の素数を与える関数とする．

$f(0) = 2,\ f(1) = 3,\ f(2) = 5,\ f(3) = 7,\ f(4) = 11,\ f(5) = 13,\ f(6) = 17,$

$f(7) = \ ?$

ℕ から ℕ への関数の例

例 1. f を自然数 n に対して，n 番目の素数を与える関数とする．

$f(0) = 2$, $f(1) = 3$, $f(2) = 5$, $f(3) = 7$, $f(4) = 11$, $f(5) = 13$, $f(6) = 17$, $f(7) = 19$

数の列は，1つの数にコードできる; 文字列も1つの数にコードできる

たとえば，f を上の関数として，$(2, 3, 10)$ を $f(0)^{2+1} \cdot f(1)^{3+1} \cdot f(2)^{10+1} = 31640625000$ に，$(0, 0, 1, 4, 2)$ を $f(0)^{0+1} \cdot f(1)^{0+1} \cdot f(2)^{1+1} \cdot f(3)^{4+1} \cdot f(4)^{2+1} = 3355517550$ に，というふうにコードすることにすれば，数を素数の積に分解する仕方は一通りしかないので，31640625000 や 3355517550 から $(2, 3, 10)$ や $(0, 0, 1, 4, 2)$ が再構成できます．

ASCII 記号表では a, b, c,... はそれぞれ 97, 98, 99,... とコードされる．

例 2. g を，自然数 n に，n を表す英語の単語のアルファベットのアスキー数の列を上のやり方でコードして得られ数を対応させる関数とする．

z, e, r, o は，アスキー記号表では，それぞれ $122, 101, 114, 111$ とコードされるので，

$$g(0) = 2^{123} \cdot 3^{102} \cdot 5^{115} \cdot 7^{112}$$

これは 260 桁の数になります！また $g(1) = 2^{112} \cdot 3^{111} \cdot 5^{102}$ で，これは 157 桁の数です！

例 3. 数学の命題を $\varphi_0, \varphi_1, \varphi_2, \ldots$ とならべておく．h を n に対し，φ_n が正しい数学の命題なら $h(n) = 1$, それ以外のときには $h(n) = 0$ とする，と定義する．

科学やテクノロジー（やその他の分野）で解決すべき問題の多くは，「あるＮからＮへの関数の計算法を求めよ」という形の問題に翻訳できる．

ＮからＮへの関数 f の計算法（アルゴリズム）とは，どんな自然数 n が与えられたときにもその方法を用いると $f(n)$ の値が求まるような具体的な手順のことです．

例1と例2での f と g には計算法があります．例2の関数 g では，計算結果は非常に大きな数になってしまいますが，それでも理論的には計算できます．

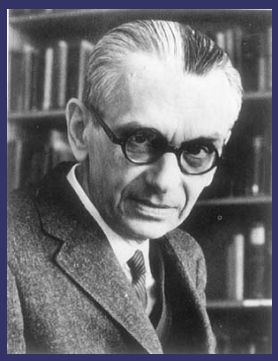
ある関数 f に計算法が存在するとき，f は計算可能である，と言うことにします．

例3の関数 h は計算可能だろうか？

もし例3の関数 h が計算可能だとすると，この計算のアルゴリズムをコンピュータにプログラムすれば，数学はコンピュータでできてしまうことになります！

ところが...

ゲーデルの不完全性定理 （1931 – 昭和6年） 例3の関数 h の計算法は存在しない．



K. ゲーデル (1906–78)

$\mathbb{N}$ から $\mathbb{N}$ へのある関数が（理論的に）計算可能でも，n を大きくすると $f(n)$ の計算に必要な手順の数が指数関数的に増えてしまう場合には，f は実際には計算不可能と言うべきでしょう．

f が計算可能な関数で，しかも，$f(n)$ の計算のための手順の数が n を大きくしたときに爆発的に大きくならないとき，f は実効的に計算可能であると言います．

n に対し，n が素数なら 1 素数でないなら 0 を返す関数は実効的に計算可能である．これに対し，n に対し n の素数分解（素数の積で表したもの）を返す関数の実効的なアルゴリズムは見つかっていない．

上の事実を使って，データを送受信するときに，データが第三者に解読できないように暗号化する方法が開発されていて，実際にいたるところで使われています．

ところが ...

n に対し n の素数分解を返す関数の実効的なアルゴリズムが存在しないことはまだ証明されていない.

この問題は $P \neq NP$ 問題と呼ばれて，理論計算機科学の最大の未解決問題と考えられています.

アメリカのマサチューセッツ州のクレイ数学研究所が懸賞を出している未解決問題のリストにもこの問題は含まれています.
（懸賞金：100万ドル）