On an aspect of the undecidable problem

Choi, Kil-Nam
Dept. of Mathematics

(Received September 30, 1985)

⟨**Abstract**⟩

This paper introduces the fact tnat cardinality of the set of computable functions is $\aleph_0$ and proves that cardinality of the set of non-computable functions is at least equal to $\aleph$ (the cardinality of continuum).

By this proof, we can see a matter of course that there are more undecidable problems than decidable problems (that is, computable ones) in idealized computer.

결정 불가능한 문제에 관하여

최 길 남
수 학 과

(1985. 9. 30 접수)

⟨요 약⟩

계산가능한 함수들의 집합의 농도가 $\aleph_0$(자연수집합의 농도)임을 주지의 사실로서 소개하고, 계산 불가능한 함수들의 집합의 농도는 적어도 $\aleph$(연속체의 농도)와 같음을 증명해 보임으로서 idealized computer에서도 결정 불가능한 문제들이 결정 가능한(즉 계산 가능한)문제보다 훨씬 더 많이 있을 수 있다는 당연함을 보인 것이다.

Ⅰ. Introduction

Over the past fifty years there have been many proposals for a precise mathematical characterisation of the intuitive idea of effective computablity. The following are some of the alternative characterisations that have been proposed;

(a) Gödel-Herbrand-Kleene (1936). General recursive functions defined by means of an equation calculus.

(b) Church(1936) λ-definable functions.

(c) Gödel-Kleene (1936) μ-recurisive functions and partial recursive functions.

(d) Turing (1936) Functions computable by finite machines known as Turing machines.

(e) Shepherdson-Sturgis (1936) URM-computable functions.

The remarkable result of investigation by many researches is the follwing.

The fundamental result 1.

Each of the above proposals for a characterisation of the notion of effective computability give rise to the same class of functions.

In view of the fundamental result 1, there claims are all mathematically equivalent. The name Church's thesis is now used to describe any of these other claims. Thus, in terms of the URM approach, we can state;

Church's thesis

The intuitively and informally defined class of effectively computable partial functions coincides exactly the class $\mathscr{C}$ of URM-computable functions.

Ⅱ. Computable functions

Our mathematical idealisation of a computer is called an unlimited register machine(URM); it is a slight variation of a machine first conceived by Shepherdson & Sturgis (1963) The URM has an infinite number of registers lablelled R_1, R_2, $R_3, \ldots$, each of which at any moment of time contains a natural number; we denote the number contained in R_n by r_n. This can be represented as follows.

R_1	R_2	R_3	R_4	R_5	R_6	R_7	
r_1	r_2	r_3	r_4	r_5	r_6	r_7	…

The contents of the registers may be altered by the URM in response to certain instructions that it can recognise.

These instructions corrsespond to very simple operations used in performing calculations with numbers. A finite list of instruction constitutes a program.

We summarise the response of the URM to the four kind of instruction in table 1.

Table 1.

Type of instruction/Instruction/Response of the URM

Zero	$Z(n)$	Replace r_n by 0 ($0 \rightarrow R_n$ or $r_n := 0$)
Sucessor	$S(n)$	Add 1 to r_n ($r_n + 1 \rightarrow R_n$, or $r_n := r_n + 1$)
Transfer	$T(m,n)$	Replace r_n by r_m ($r_m \rightarrow R_n$ or $r_n := r_m$)
Jump	$J(m,n,q)$	If $r_m = r_n$, jump to the qth instruction; otherwise go on to the next instruction in the program

Definitions 1.

Let f be a partial function from N^n to N. ($N = \{0, 1, 2, \cdots\}$)

(a) Suppose that P is program, and let $a_1, a_2 \cdots a_n$, $b \in N$.

(i) The computation $P\ (a_1, a_2, \cdots, a_n)$ converges to b if $P(a_1, a_2, \cdots a_n)\downarrow$ and in the final configuration b is in R_1.

(ii) P URM-computes f if, for every a_1, $a_2, \cdots, a_n$, b $P(a_1, \cdots a_n)\downarrow b$ if and only if $(a_1, \cdots, a_n) \in \mathrm{Dom}\ f$ and $f\ (a_1, \cdots, a_n) = b$

(b) The funtion f is URM-computable if there is a program that URM-computes f.

The class of URM-computable functions is denoted by $\mathscr{C}$, and n-ary URM-computable functions by $\mathscr{C}_n$.

Suppose that $M(x_1,\ x_2, \cdots, x_n)$ is an n-ary predicate of natural numbers.

The characteristic function $C_M(x)$ (letting $x = (x_1, x_2, \cdots, x_n)$ is given by

$$C_M(x) = \begin{cases} 1 & \text{if } M(x) \text{ holds} \\ 0 & \text{if } M(x) \text{ doesn't hold.} \end{cases}$$

Definition 2.

The predicate $M(x)$ is decidable if the function C_M is computable.

$M(x)$ is undecidable if $M(x)$ is not decidable.

Let us now denote the set of all URM instruction by $\mathscr{I}$, and the set of all program by $\mathscr{P}$, A program consists of a finite list of instructions.

Theorem 1.

$\mathscr{I}$ is effectively denumerable.

Proof. We define an explicit bijection α: $\mathscr{I} \rightarrow N$ that maps the four kind of instruction onto natural numbers the forms $4u$, $4u+1$, $4u+2$, $4u+3$ respectively;

$$\alpha(Z(n)) = 4n$$
$$\alpha(S(n)) = 4n+1$$
$$\alpha(T(m,n)) = 4 \times 2^m \times 3^n + 2$$
$$\alpha(J(m,n,q)) = 4 \times 2^m \times 3^n \times 5^q + 3$$

This explicit definition shows that α is effectively computable. To find $\alpha^{-1}(x)$, find u, r such that $x=4u+r$ with $0\leqq r<4$.

The value of r indicates which kind of instruction $\alpha^{-1}(x)$ is, and from u we can effectively find the particular instruction. Hence α^{-1} is also effectively computable.

Corollary 1.

$\mathscr{P}$ is effectively denumerable.

Proof. The finite union of denumerable sets is denumerable.

The cardinal number of the set of computable partial functions is $\aleph_0$.

The following theorem is then well known (see Cutland).

Theorem 2 (The *s-m-n* theorem).

Suppose that $f(x,y)$ is computable function.

There is a total computable function $k(x)$ such that

$$f(x,y)=\phi_{k(x)}(y).$$

Ⅲ. Undecidable problems in computability

Definition 3.

For each $a\in N$, and $n\geqq1$;

(a) $\phi_a^{(n)}$=the n-ary function computed by P_a

(b) $W_a^{(n)}$=domain of $\phi_a^{(n)}=\{(x_1,\cdots,x_n):P_a(x_1,\cdots x_n)\downarrow\}$, $E_a^{(n)}$=range of $\phi_a^{(n)}$; $\phi_a^{(1)}$, $W_a^{(1)}$ and $E_a^{(1)}$ are denoted by ϕ_a, W_a and E_a.

Theorem 3.

'$x\in W_x$'(or, equivalently, '$\phi_x(x)$ is defined', or '$P_x(x)\downarrow$') is undecidable.

Proof. The characteristic function f of this problem is given by

$$f(x)=\begin{cases}1 \text{ if } x\in W_x\\0 \text{ if } x\notin W_x.\end{cases}$$

Suppose that f is computable; we shall obtain a contradiction. Specifically, we make a diagonal construction of a computable function g such that $\mathrm{Dom}(g)\neq W_x(=\mathrm{Dom}(\phi_x))$ for every x;

This is obviously contradictory.

The diagonal motto tells us to ensure that $\mathrm{Dom}(g)$ differs from W_x at x; so we aim to make

$$x\in\mathrm{Dom}(g)\Longleftrightarrow x\notin W_x.$$

Let us define g, then, by

$$g(x)=\begin{cases}0 & \text{if } x\notin W_x \text{ (i.e. if } f(x)=0)\\ \text{undefined} & \text{if } x\in W_x \text{ (i.e. if } f(x)=1)\end{cases}$$

Since f is computable, then so is g (by Church's thesis); So we have our contradiction. Therefore, we can conclude that f is not computable, and so the problem '$x\in W_x$' is undecidable.

The following theorems are well known (see Cutland).

Theorem 4. (the Halting problem)

The problem '$\phi_x(y)$ is defined'(or, equivalently '$P_x(y)$' or '$y\in W_x$') is undecidable.

Theorem 5.

Let c be any number. The following problems are undecidable.

(a) (the Input or Acceptance problem) '$c\in W_x$' (equivalently, '$P_x(c)\downarrow$' or $c\in\mathrm{Dom}(\phi_x)$').

(b) (the Output or Printing problem) '$c\in E_x$' (equivalently, $c\in\mathrm{Ran}(\phi_x)$').

Ⅳ. Main Results

Theorem 6.

The cardinality of the set of all partial functions from N to N is at least $\aleph$.

Proof. Let $F=\{f|f:N\to N\}$ and $G=\{f|f:N\to\{0,1\}\}$.

Since Card $G=2^N=\aleph$, Card $F\geqq\aleph$.

Corollary 2

The set of all non computable functions from N to N is nondenumerable.

Proof. Suppose that the set of all non-computable function is denumerable. Since the set of all computable functions is denumerable and the union of two denumerable sets is denumerable, this is a contradiction. Therefore, the set of all noncomputable functions is nondenumerable. That is, it shows that there are more undicidable problems that decidable problems.

References

1. Cutland, N. Computability; an introduction to recursive function theory, Cambridge university press, Cambridge, 1980.
2. Rogers, H. Theory of recursive functions and effective computability, McGraw-Hill, New York, 1967.