Computability Theory VII

S-M-N Theorem

Qingshui Xue

Shanghai Jiao Tong University

Nov. 2, 2015

Problem Index

Motivation

By Church-Turing Thesis one may study computability theory using any of the computation models.

It is much more instructive however to carry out the study in a model independent manner.

The first step is to assign index to computable function.

Review Tips

Effective Denumerable Set

$\mathbb{N} \times \mathbb{N}$

$\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$

$\bigcup_{k>0} \mathbb{N}^k$

Effective Denumerable Set

$\mathbb{N} \times \mathbb{N}$

$\mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+$

$\bigcup_{k>0} \mathbb{N}^k$

$\mathcal{I}$ is effectively denumerable.

$\mathcal{P}$ is effectively denumerable.

$\gamma(P) = \tau(\beta(I_1), \ldots, \beta(I_s))$

The value $\gamma(P)$ is called the Gödel number of P.

Synopsis

1. Gödel Index
2. S-m-n Theorem

GÖDEL INDEX

Basic Idea

We see a number as an index for a problem/function if it is the Gödel number of a programme that solves/calculates the problem/function.

Definition

Suppose $a \in \mathbb{N}$ and $n \geq 1$.

$$
\begin{aligned}
\phi_a^{(n)} &= \text{the } n \text{ ary function computed by } P_a \\
&= f_{P_a}^{(n)}, \\
W_a^{(n)} &= \text{the domain of } \phi_a^{(n)} = \{(x_1, \ldots, x_n) \mid P_a(x_1, \ldots, x_n) \downarrow\}, \\
E_a^{(n)} &= \text{the range of } \phi_a^{(n)}.
\end{aligned}
$$

The super script (n) is omitted when $n = 1$.

Example

Let $a = 4127$. Then $P_{4127} = S(2); T(2,1)$.

Example

Let $a = 4127$. Then $P_{4127} = S(2); T(2,1)$.

If the program is seen to calculate a unary function, then

$$
\begin{aligned}
\phi_{4127}(x) &= 1, \\
W_{4127} &= \mathbb{N}, \\
E_{4127} &= \{1\}.
\end{aligned}
$$

Example

Let $a = 4127$. Then $P_{4127} = S(2); T(2,1)$.

If the program is seen to calculate a unary function, then

$$\begin{aligned} \phi_{4127}(x) &= 1, \\ W_{4127} &= \mathbb{N}, \\ E_{4127} &= \{1\}. \end{aligned}$$

If the program is seen to calculate an n-ary function, then

$$\begin{aligned} \phi^{(n)}_{4127}(x_1, \ldots, x_n) &= x_2 + 1, \\ W^n_{4127} &= \mathbb{N}^n, \\ E^n_{4127} &= \mathbb{N}^+. \end{aligned}$$

Gödel Index for Computable Function

Suppose f is an n-ary computable function..

A number a is an index for f if $f = \phi_a^{(n)}$.

Padding Lemma

Padding Lemma

Every computable function has infinite indices. Moreover for each x we can effectively find an infinite recursive set A_x of indices for ϕ_x.

Padding Lemma

Padding Lemma

Every computable function has infinite indices. Moreover for each x we can effectively find an infinite recursive set A_x of indices for ϕ_x.

Proof

Systematically add useless instructions to P_x.

Enumeration of Computable Function

Proposition

$\mathcal{C}_n$, and $\mathcal{C}$ as well, is denumerable.

Enumeration of Computable Function

Proposition

$\mathcal{C}_n$, and $\mathcal{C}$ as well, is denumerable.

We may list for example all the elements of $\mathcal{C}_n$ as $\phi_0^{(n)}, \phi_1^{(n)}, \phi_2^{(n)}, \ldots$.

Diagonal Method

Fact

There is a total unary function that is not computable.

Diagonal Method

Fact

There is a total unary function that is not computable.

Proof

Suppose $\phi_0, \phi_1, \phi_2, \ldots$ is an enumeration of $\mathcal{C}$. Define

$$f(n) = \begin{cases} \phi_n(n) + 1, & \text{if } \phi_n(n) \text{ is defined,} \\ 0, & \text{if } \phi_n(n) \text{ is undefined.} \end{cases}$$

By Church-Turing Thesis the function $f(n)$ is not computable.

Diagonal Method

Fact

There is a total unary function that is not computable.

Proof

Suppose $\phi_0, \phi_1, \phi_2, \ldots$ is an enumeration of $\mathcal{C}$. Define

$$f(n) \quad = \quad \begin{cases} \phi_n(n) + 1, & \text{if } \phi_n(n) \text{ is defined,} \\ 0, & \text{if } \phi_n(n) \text{ is undefined.} \end{cases}$$

By Church-Turing Thesis the function $f(n)$ is not computable.

Is the following function computable?

$$f(n) \quad \simeq \quad \begin{cases} \phi_n(n) + 1, & \text{if } \phi_n(n) \text{ is defined,} \\ \uparrow, & \text{if } \phi_n(n) \text{ is undefined.} \end{cases}$$

Diagonal Method

Suppose there is a sequence $f_0, f_1, \ldots, f_n, \ldots$

Diagonalize out of $f_0, f_1, \ldots$ by making f differ from f_n at n.

S-m-n Theorem

Motivation

How do different indexing systems relate?

S-m-n Theorem, the Unary Case

Given a binary function $f(x, y)$, we get a unary computable function $f(a, y)$ by fixing a value a for x.

S-m-n Theorem, the Unary Case

Given a binary function $f(x, y)$, we get a unary computable function $f(a, y)$ by fixing a value a for x.

Let e be an index for $f(a, y)$. Then

$$f(a, y) \simeq \phi_e(y)$$

S-m-n Theorem, the Unary Case

Given a binary function $f(x, y)$, we get a unary computable function $f(a, y)$ by fixing a value a for x.

Let e be an index for $f(a, y)$. Then

$$f(a, y) \simeq \phi_e(y)$$

S-m-n Theorem states that the index e can be computed from a.

S-m-n Theorem, the Unary Case

Fact

Suppose that $f(x, y)$ is a computable function. There is a primitive recursive function $k(x)$ such that

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

S-m-n Theorem, the Unary Case

Proof

S-m-n Theorem, the Unary Case

Proof

Let F be a program that computes f. Consider the following program

$$\begin{array}{l} T(1,2) \\ Z(1) \\ \left.\begin{array}{l} S(1) \\ \ \vdots \\ S(1) \end{array}\right\} a \text{ times} \\ F \end{array}$$

S-m-n Theorem, the Unary Case

Proof

Let F be a program that computes f. Consider the following program

$$\begin{array}{l} T(1,2) \\ Z(1) \\ \left.\begin{array}{l} S(1) \\ \vdots \\ S(1) \end{array}\right\} a \text{ times} \\ F \end{array}$$

The above program can be effectively constructed from a.

S-m-n Theorem, the Unary Case

Proof

Let F be a program that computes f. Consider the following program

$$
\begin{array}{l}
T(1,2) \\
Z(1) \\
\left.\begin{array}{l} S(1) \\ \vdots \\ S(1) \end{array}\right\} a \text{ times} \\
F
\end{array}
$$

The above program can be effectively constructed from a.

Let $k(a)$ be the Gödel number of the above program. It can be effectively computed from the above program.

Examples

Let $f(x, y) = y^x$.

Examples

Let $f(x, y) = y^x$.

Then $\phi_{k(x)}(y) = y^x$. For each fixed n, $k(n)$ is an index for y^n.

Examples

Let $f(x, y) = y^x$.

Then $\phi_{k(x)}(y) = y^x$. For each fixed n, $k(n)$ is an index for y^n.

Let $f(x, y) \simeq \begin{cases} y, & \text{if } y \text{ is a multiple of } x, \\ \uparrow, & \text{otherwise.} \end{cases}$.

Examples

Let $f(x, y) = y^x$.

Then $\phi_{k(x)}(y) = y^x$. For each fixed n, $k(n)$ is an index for y^n.

Let $f(x, y) \simeq \begin{cases} y, & \text{if } y \text{ is a multiple of } x, \\ \uparrow, & \text{otherwise.} \end{cases}$.

Then $\phi_{k(n)}(y)$ is defined if and only if y is a multiple of n.

S-m-n Theorem

S-m-n Theorem

For m, n, there is an injective primitive recursive $(m+1)$-function $s_n^m(x, \widetilde{x})$ such that for all e the following holds:

$$\phi_e^{(m+n)}(\widetilde{x}, \widetilde{y}) \simeq \phi_{s_n^m(e,\widetilde{x})}^{(n)}(\widetilde{y})$$

S-m-n Theorem

S-m-n Theorem

For m, n, there is an injective primitive recursive $(m+1)$-function $s_n^m(x, \widetilde{x})$ such that for all e the following holds:

$$\phi_e^{(m+n)}(\widetilde{x}, \widetilde{y}) \simeq \phi^{(n)}_{s_n^m(e,\widetilde{x})}(\widetilde{y})$$

S-m-n Theorem is also called **Parameter Theorem**.

S-m-n Theorem

Proof

Given $e, x_1, \ldots, x_m$, we can effectively construct the following program and its index

$$\begin{array}{l} T(n, m+n) \\ \vdots \\ T(1, m+1) \\ Q(1, x_1) \\ \vdots \\ Q(m, x_m) \\ P_e \end{array}$$

where $Q(i, x)$ is the program $Z(i), \underbrace{S(i), \ldots, S(i)}_{x \text{ times}}$.

S-m-n Theorem

Proof

Given $e, x_1, \ldots, x_m$, we can effectively construct the following program and its index

$$\begin{array}{l} T(n, m+n) \\ \vdots \\ T(1, m+1) \\ Q(1, x_1) \\ \vdots \\ Q(m, x_m) \\ P_e \end{array}$$

where $Q(i, x)$ is the program $Z(i), \underbrace{S(i), \ldots, S(i)}_{x \text{ times}}$.

The injectivity is achieved by padding enough useless instructions.

Exercise I

Show that there is a total computable function k such that for each n, $k(n)$ is an index of the function $\lceil \sqrt[n]{x} \rceil$.

Exercise II

Show that there is a total computable function k such that for each n, $W_{k(n)} =$ the set of perfect nth power.

Exercise III

Show that there is a total computable function k such that

$$W_{k(n)}^{(m)} = \{(y_1, \ldots, y_m) : y_1 + y_2 + \ldots + y_m = n\}$$

suppose $m \geq 1$.