

Lecture 8: Theorems on Computability

Lecturer: Renjie Yang

8.1 Theorems on Computability

Definition 8.1 We write $f(x) \simeq g(x)$ to mean that either $f(x)$ and $g(x)$ are both undefined, or they are both defined and equal.

Theorem 8.2 (Kleene's Normal Form Theorem) For any natural number $n > 0$, there are a primitive computable relation $T(n, x^{(n)}, y)$ and a primitive recursive function U such that for each partial computable (partial recursive) function $f(x^{(n)})$ there is a number z_0 , such that

$$f(x^{(n)}) \simeq U(\min_y T_n(z_0, x^{(n)}, y))$$

Proof: This is Corollary 7.11 written in a slightly more general form. ■

Theorem 8.3 (Effective Enumeration) For any natural number $n > 0$, there is a universal partial computable function $Un(k, x^{(n)})$ such that

1. $Un(k, x^{(n)})$ is partial computable.
2. If $f(x^{(n)})$ is any partial computable function, then there is a natural number k such that $f(x^{(n)}) \simeq Un(k, x^{(n)})$ for every sequence $x^{(n)} = (x_0, x_1, \dots, x_{n-1})$.

Proof: Let $Un(k, x^{(n)}) \simeq U(\min_y T_n(k, x^{(n)}, y))$ in Theorem 8.2. ■

Definition 8.4 Let φ_i denotes the i th unary partial computable function $Un(i, x)$. Use φ_i^n to denote the i th n -ary partial recursive function $Un(i, x^{(n)})$.

Theorem 8.5 For any natural number $n > 0$, there is a partial computable function $f(i, x^{(n)})$ such that for each i and n and a sequence of numbers a_0, a_1, \dots, a_{n-1} , we have

$$f(i, a_0, a_1, \dots, a_{n-1}) \simeq \varphi_i^n(a_0, a_1, \dots, a_{n-1})$$

Proof: Let $f(i, x^{(n)}) = Un(i, x^{(n)})$, the conclusion holds by the definition of φ_i^n . ■

Note While Theorem 8.5 defines a universal computable function for each k , it is also possible to define a binary function $g(i, x)$ which treats its second argument as the Gödel code for a finite sequence (x_0, \dots, x_{n-1}) , and then computes in the same manner as the n -ary universal function so that we have $g(i, gn(x_0, \dots, x_{n-1})) \simeq f(i, x_0, \dots, x_{n-1})$. This provides a means of replacing the prior enumerations of any arbitrary n -ary partial computable functions with a single enumeration $\phi_i(x)$ of unary functions, such that $\phi_i(gn(x_0, \dots, x_{n-1})) \simeq f(i, gn(x_0, \dots, x_{n-1})) \simeq \varphi_i^n(x_0, \dots, x_{n-1})$. Thus the following theorem:

Theorem 8.6 (Universal Computable Function) There is a partial computable function $g(x, y)$ such that for each i and n and sequence of numbers a_0, \dots, a_{n-1} we have

$$g(i, gn(a_0, \dots, a_{n-1})) \simeq \varphi_i^n(a_0, \dots, a_{n-1})$$

Proof: Let $g(i, x) \simeq Un(i, x)$, and define $Un(i, gn(a_0, \dots, a_{n-1})) \simeq \varphi_i^n(a_0, \dots, a_{n-1})$. ■

Theorem 8.7 (*s-m-n*) *For each pair of natural numbers n and m , there is a primitive recursive function s_n^m such that for every sequence $x, a_0, \dots, a_{m-1}, y_0, \dots, y_{n-1}$, we have*

$$\varphi_{s_n^m(x, a_0, \dots, a_{m-1})}^n(y_0, \dots, y_{n-1}) \simeq \varphi_x^{m+n}(a_0, \dots, a_{m-1}, y_0, \dots, y_{n-1})$$

8.2 The Church-Turing Thesis

The Church-Turing thesis concerns the concept of an effective or systematic or mechanical method in logic, mathematics and computer science.

“Effective” and its synonyms “systematic” and “mechanical” are terms of art in these disciplines: they do not carry their everyday meaning. A method, or procedure, M , for achieving some desired result is called “effective” (or “systematic” or “mechanical”) just in case:

1. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
2. M will, if carried out without error, produce the desired result in a finite number of steps;
3. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
4. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Turing’s Thesis: L.C.M.s [logical computing machines: Turing’s expression for Turing machines] can do anything that could be described as “rule of thumb” or “purely mechanical”.

Church’s Thesis: A function of positive integers is effectively calculable only if lambda-definable (or, equivalently, recursive).

Further reading: The SEP Entry: “The Church-Turing Thesis”.