Chapter 5

Recursion Theory; More Advanced Topics

5.1 The Recursion Theorem

The recursion Theorem, due to Kleene, is a fundamental result in recursion theory.

Theorem 5.1. *(Recursion Theorem, Version 1) Let $\varphi_0, \varphi_1, \ldots$ be any acceptable indexing of the partial computable functions. For every total computable function f, there is some n such that*

$$\varphi_n = \varphi_{f(n)}.$$

The recursion Theorem can be strengthened as follows.

Theorem 5.2. *(Recursion Theorem, Version 2) Let $\varphi_0, \varphi_1, \ldots$ be any acceptable indexing of the partial computable functions. There is a total computable function h such that for all $x \in \mathbb{N}$, if φ_x is total, then*

$$\varphi_{\varphi_x(h(x))} = \varphi_{h(x)}.$$

A third version of the recursion Theorem is given below.

Theorem 5.3. *(Recursion Theorem, Version 3) For all $n \geq 1$, there is a total computable function h of $n+1$ arguments, such that for all $x \in \mathbb{N}$, if φ_x is a total computable function of $n+1$ arguments, then*

$$\varphi_{\varphi_x(h(x,x_1,\ldots,x_n),x_1,\ldots,x_n)} = \varphi_{h(x,x_1,\ldots,x_n)},$$

for all $x_1, \ldots, x_n \in \mathbb{N}$.

As a first application of the recursion theorem, we can show that there is an index n such that φ_n is the constant function with output n.

Loosely speaking, φ_n prints its own name. Let f be the computable function such that

$$f(x, y) = x$$

for all $x, y \in \mathbb{N}$.

By the s-m-n Theorem, there is a computable function g such that

$$\varphi_{g(x)}(y) = f(x, y) = x$$

for all $x, y \in \mathbb{N}$.

By the recursion Theorem 5.1, there is some n such that

$$\varphi_{g(n)} = \varphi_n,$$

the constant function with value n.

As a second application, we get a very short proof of Rice's Theorem.

Let C be such that $P_C \neq \emptyset$ and $P_C \neq \mathbb{N}$, and let $j \in P_C$ and $k \in \mathbb{N} - P_C$.

Define the function f as follows:

$$f(x) = \begin{cases} j & \text{if } x \notin P_C, \\ k & \text{if } x \in P_C, \end{cases}$$

If P_C is computable, then f is computable. By the recursion Theorem 5.1, there is some n such that

$$\varphi_{f(n)} = \varphi_n.$$

But then, we have

$$n \in P_C \quad \text{iff} \quad f(n) \notin P_C$$

by definition of f, and thus,

$$\varphi_{f(n)} \neq \varphi_n,$$

a contradiction.

Hence, P_C is not computable.

As a third application, we have the following Lemma.

Lemma 5.4. *Let C be a set of partial computable functions and let*

$$A = \{x \in \mathbb{N} \mid \varphi_x \in C\}.$$

The set A is not reducible to its complement $\overline{A}$.

The recursion Theorem can also be used to show that functions defined by recursive definitions other than primitive recursion are partial computable.

This is the case for the function known as *Ackermann's function*, defined recursively as follows:

$$\begin{aligned} f(0, y) &= y + 1, \\ f(x + 1, 0) &= f(x, 1), \\ f(x + 1, y + 1) &= f(x, f(x + 1, y)). \end{aligned}$$

It can be shown that this function is not primitive recursive. Intuitively, it outgrows all primitive recursive functions.

However, f is computable, but this is not so obvious.

We can use the recursion Theorem to prove that f is computable. Consider the following definition by cases:

$$\begin{aligned} g(n,0,y) &= y+1, \\ g(n,x+1,0) &= \varphi_{univ}(n,x,1), \\ g(n,x+1,y+1) &= \varphi_{univ}(n,x,\varphi_{univ}(n,x+1,y)). \end{aligned}$$

Clearly, g is partial computable. By the s-m-n Theorem, there is a computable function h such that

$$\varphi_{h(n)}(x,y) = g(n,x,y).$$

By the recursion Theorem, there is an m such that

$$\varphi_{h(m)} = \varphi_m.$$

Therefore, the partial computable function $\varphi_m(x,y)$ satisfies the definition of Ackermann's function.

We showed in a previous Section that $\varphi_m(x, y)$ is a total function, and thus, Ackermann's function is a total computable function.

Hence, the recursion Theorem justifies the use of certain recursive definitions. However, note that there are some recursive definition that are only satisfied by the completely undefined function.

In the next Section, we prove the extended Rice Theorem.

5.2 Extended Rice Theorem

The extended Rice Theorem characterizes the sets of partial computable functions C such that P_C is c.e.

First, we need to discuss a way of indexing the partial computable functions that have a finite domain.

Using the uniform projection function Π, we define the primitive recursive function F such that

$$F(x, y) = \Pi(y + 1, \Pi_1(x) + 1, \Pi_2(x)).$$

We also define the sequence of partial functions $P_0, P_1, \ldots$ as follows:

$$P_x(y) = \begin{cases} F(x, y) - 1 & \text{if } 0 < F(x, y) \text{ and } y < \Pi_1(x) + 1, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Lemma 5.5. *Every P_x is a partial computable function with finite domain, and every partial computable function with finite domain is equal to some P_x.*

The easy part of the extended Rice Theorem is the following lemma.

Recall that given any two partial functions $f\colon A \to B$ and $g\colon A \to B$, we say that *g extends f* iff $f \subseteq g$, which means that $g(x)$ is defined whenever $f(x)$ is defined, and if so, $g(x) = f(x)$.

Lemma 5.6. *Let C be a set of partial computable functions. If there is an c.e. set A such that, $\varphi_x \in C$ iff there is some $y \in A$ such that φ_x extends P_y, then $P_C = \{x \mid \varphi_x \in C\}$ is c.e.*

Proof. Lemma 5.6 can be restated as

$$P_C = \{x \mid \exists y \in A,\ P_y \subseteq \varphi_x\}.$$

If A is empty, so is P_C, and P_C is c.e.

Otherwise, let f be a computable function such that

$$A = range(f).$$

Let ψ be the following partial computable function:

$$\psi(z) = \begin{cases} \Pi_1(z) & \text{if } P_{f(\Pi_2(z))} \subseteq \varphi_{\Pi_1(z)}, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

It is clear that

$$P_C = range(\psi).$$

To see that ψ is partial computable, write $\psi(z)$ as follows:

$$\psi(z) = \begin{cases} \Pi_1(z) & \text{if } \forall w \leq \Pi_1(f(\Pi_2(z))) \\ & \quad [F(f(\Pi_2(z)), w) > 0 \supset \\ & \quad \varphi_{\Pi_1(z)}(w) = F(f(\Pi_2(z)), w) - 1], \\ \text{undefined} & \text{otherwise.} \end{cases}$$

$\square$

To establish the converse of Lemma 5.6, we need two Lemmas.

Lemma 5.7. *If P_C is c.e. and $\varphi \in C$, then there is some $P_y \subseteq \varphi$ such that $P_y \in C$.*

As a corollary of Lemma 5.7, we note that TOTAL is not c.e.

Lemma 5.8. *If P_C is c.e., $\varphi \in C$, and $\varphi \subseteq \psi$, where ψ is a partial computable function, then $\psi \in C$.*

Observe that Lemma 5.8 yields a new proof that $\overline{\text{TOTAL}}$ is not c.e. Finally, we can prove the extended Rice Theorem.

Theorem 5.9. *(Extended Rice Theorem) The set P_C is c.e. iff there is a c.e. set A such that*

$$\varphi_x \in C \quad \textit{iff} \quad \exists y \in A\, (P_y \subseteq \varphi_x).$$

Proof. Let $P_C = dom(\varphi_i)$. Using the s-m-n Theorem, there is a computable function k such that

$$\varphi_{k(y)} = P_y$$

for all $y \in \mathbb{N}$.

Define the c.e. set A such that

$$A = dom(\varphi_i \circ k).$$

Then,

$$y \in A \quad \text{iff} \quad \varphi_i(k(y)) \downarrow \quad \text{iff} \quad P_y \in C.$$

Next, using Lemma 5.7 and Lemma 5.8, it is easy to see that

$$\varphi_x \in C \quad \text{iff} \quad \exists y \in A \, (P_y \subseteq \varphi_x).$$

$\square$

5.3 Creative and Productive Sets; Incompleteness

In this section, we discuss some special sets that have important applications in logic: *creative and productive sets*.

The concepts to be described are illustrated by the following situation. Assume that

$$W_x \subseteq \overline{K}$$

for some $x \in \mathbb{N}$.

We claim that

$$x \in \overline{K} - W_x.$$

Indeed, if $x \in W_x$, then $\varphi_x(x)$ is defined, and by definition of K, we get $x \notin \overline{K}$, a contradiction.

Therefore, $\varphi_x(x)$ must be undefined, that is,

$$x \in \overline{K} - W_x.$$

The above situation can be generalized as follows.

Definition 5.1. A set A is *productive* iff there is a total computable function f such that

$$\text{if} \quad W_x \subseteq A \quad \text{then} \quad f(x) \in A - W_x$$

for all $x \in \mathbb{N}$. The function f is called the *productive function of A*. A set A is *creative* if it is c.e. and if its complement $\overline{A}$ is productive.

As we just showed, K is creative and $\overline{K}$ is productive. The following facts are immediate conequences of the definition.

(1) A productive set is not c.e.

(2) A creative set is not computable.

Creative and productive sets arise in logic.

The set of theorems of a logical theory is often creative. For example, the set of theorems in Peano's arithmetic is creative. This yields incompleteness results.

Lemma 5.10. *If a set A is productive, then it has an infinite c.e. subset.*

Another important property of productive sets is the following.

Lemma 5.11. *If a set A is productive, then $\overline{K} \leq A$.*

Using Lemma 5.11, the following results can be shown.

Lemma 5.12. *The following facts hold.*

(1) If A is productive and $A \leq B$, then B is productive.

(2) A is creative iff A is equivalent to K.

(3) A is creative iff A is complete,