THEORY OF COMPUTATION
Recursively Enumerable Sets - 14 part 5

Prof. Dan A. Simovici

UMB

The purpose of Rice's Theorem is to provide a tool that allows us to prove that certain general sets are not recursive.

Definition

Let Γ be a collection of partially computable functions of one variable. The index set associated with Γ is the set

$$R_\Gamma = \{t \in \mathbb{N} \mid \Phi_t \in \Gamma\}.$$

An index set R_Γ contains program codes that display a certain input-output behavior as specified by Γ.

By Church's Thesis, R_Γ is a recursive set when there is an algorithm that accepts program codes as inputs and returns the value TRUE or FALSE depending on whether or not the function $\psi_{\mathcal{P}}^{(1)}$ does or does not belong to Γ.

Examples of sets of functions Γ:

- the set of computable functions;
- the set of primitive recursive functions;
- the set of partially computable functions that are defined for all but a finite numbers of values of x.

It would be pleasing to have algorithms that accept a program as input and return as output some useful property of the partial function computed by the program. Unfortunately, such algorithms do not exist.

Theorem

Rice's Theorem: *Let Γ be a collection of partially computable functions of one variable. Let there be partially computable functions $f(x)$ and $g(x)$ such that $f(x)$ belongs to Γ and $g(x)$ does not belong to Γ. Then R_Γ is not recursive.*

Proof.

Let $h(x)$ be the function such that $h(x)\uparrow$ for all x, that is the empty function. There are two cases to discuss:

- $h(x) \notin \Gamma$, and
- $h(x) \in \Gamma$.

Suppose initially that $h(x) \notin \Gamma$

Let q be the number of the program

$$Z \leftarrow \Phi(X_2, X_2)$$
$$Y \leftarrow f(X_1)$$

Let $\Phi(x_1, x_2, q)$ be the function computed by this program. By the smn Theorem we have $\Phi(x_1, x_2, q) = \Phi(x_1, S^1_1(x_2, q))$. □

Proof cont'd

Proof.

Thus, $S_1^1(i, q)$ is the number of the program

$$X_2 \leftarrow i$$
$$Z \leftarrow \Phi(X_2, X_2)$$
$$Y \leftarrow f(X_1)$$

that computes the function f. □

Proof cont'd

Proof.

Note that

$$\begin{aligned} i \in K &\Rightarrow \Phi(i,i)\downarrow \\ &\Rightarrow \Phi_{S_1^1(i,q)}(x) = f(x) \text{ for all } x \\ &\Rightarrow \Phi_{S_1^1(i,q)} \in \Gamma \\ &\Rightarrow S_1^1(i,q) \in R_\Gamma . \end{aligned}$$

□

Proof cont'd

Also,

Proof.

$$\begin{aligned} i \notin K &\Rightarrow \Phi(i,i)\uparrow \\ &\Rightarrow \Phi_{S_1^1(i,q)}(x)\uparrow \text{ for all } x \\ &\Rightarrow \Phi_{S_1^1(i,q)} = h \\ &\Rightarrow \Phi_{S_1^1(i,q)} \notin \Gamma \\ &\Rightarrow S_1^1(i,q) \notin R_\Gamma, \end{aligned}$$

so $K \leqslant_m R_\Gamma$. Therefore, R_Γ is not recursive. □

Proof cont'd

Proof.

If $h(x)$ does not belong to Γ, the same argument with Γ and $f(x)$ replaced by $\overline{\Gamma}$ and $g(x)$, respectively, shows that $R_{\overline{\Gamma}}$ is not recursive.
Since $R_{\overline{\Gamma}} = \overline{R_\Gamma}$, R_Γ is not recursive in this case either. □

Corollary

There are no algorithms for testing a given program $\mathcal{P}$ of the language $\mathcal{S}$ to determine whether $\psi_{\mathcal{P}}^{(1)}(x)$ belongs to any of the classes:

- *the set of primitive recursive functions;*
- *the set of partially computable functions that are defined for all but a finite numbers of values of x.*

Proof.

In each case we need to find the required functions f and g to show that R_Γ is not recursive. For example, the functions $f(x) = x$ and $g(x) = 1 - x$ (so that g is defined only for $x = 0$ or $x = 1$) work. □

The Other Rice's Theorem offers a technique for proving that certain sets are nor r.e.

Definition

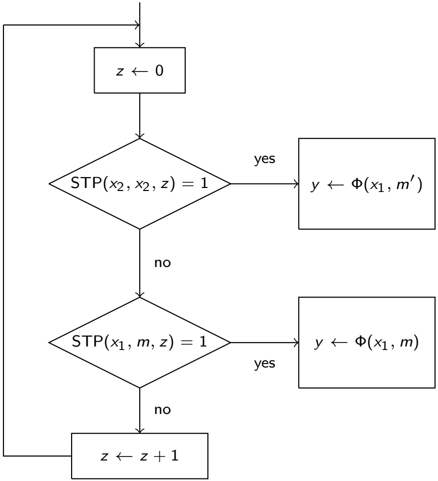
Let f, g be two partial functions. We write $f \subseteq g$ if $x \in \text{Dom}(f)$ implies $x \in \text{Dom}(g)$ and $g(x) = f(x)$.

Theorem

The Other Rice's Theorem: *Let* Γ *be a set of computable functions. If there exist* m, m' *such that* $\Phi_m \in \Gamma$, $\Phi_{m'} \notin \Gamma$ *and* $\Phi_m \subseteq \Phi_{m'}$, *then* R_Γ *is not r.e..*

Proof

Consider the flowchart shown on the next slide that can be readily transformed into a $\mathcal{S}$ program $\mathcal{P}$, where $\#\mathcal{P} = p$. Note that the execution of this program depends on the input value x_2.


$z \leftarrow 0$
$STP(x_2, x_2, z) = 1$
yes
$y \leftarrow \Phi(x_1, m')$
no
$STP(x_1, m, z) = 1$
yes
$y \leftarrow \Phi(x_1, m)$
no
$z \leftarrow z + 1$

Proof cont'd

The equivalent program $\mathcal{P}$ with $\#(\mathcal{P}) = p$ can be written as:

$$
\begin{array}{ll}
 & Z \leftarrow 0 \\
[C] & \text{IF STP}(X_2, X_2, Z) \text{ GOTO } A \\
 & \text{IF STP}(X_1, m, Z) \text{ GOTO } B \\
 & Z \leftarrow Z + 1 \\
 & \text{GOTO } C \\
[A] & Y \leftarrow \Phi(X_1, m) \\
 & \text{GOTO } E \\
[B] & Y \leftarrow \Phi(X_1, m')
\end{array}
$$

Proof cont'd

If $x_2 \in K$, we have $\Phi(x_1, x_2, p) = \Phi_{m'}(x_1)$; if $x_2 \notin K$, then $\Phi(x_1, x_2, p) = \Phi_m(x_1)$.
Thus, we have:

$$\Phi(x_1, x_2, p) = \begin{cases} \Phi_m(x_1) & \text{if } x_2 \notin K \\ \Phi_{m'}(x_1) & \text{if } x_2 \in K. \end{cases}$$

By the smn Theorem we have:

$$\Phi_{S_1^1(x_2,p)}(x_1) = \begin{cases} \Phi_m(x_1) & \text{if } x_2 \notin K (\text{that is, } x_2 \in \overline{K}) \\ \Phi_{m'}(x_1) & \text{if } x_2 \in K. \end{cases}$$

Define f as $f(x_2) = S_1^1(x_2, p)$. We have:

- $x \in \overline{K}$ if and only if $\Phi_{f(x_2)} = \Phi_m$, that is, $f(x_2) \in R_\Gamma$;
- $x \in K$ if and only if $\Phi_{f(x_2)} = \Phi_{m'}$, that is, $f(x_2) \notin R_\Gamma$.

Thus, $x \in \overline{K}$ if and only if $f(x) \in \Gamma$, so $\overline{K} \leqslant R_\Gamma$, which implies that R_Γ is not r.e.

Example

The following sets can be shown not to be r.e. using the Other Rice's Theorem:

- $\text{EMPTY} = \{x \mid \text{Dom}(\Phi_x) = \emptyset\}$;
- $\{x \mid \textbf{range}(\Phi_x) = \emptyset\}$;
- $\text{FIN} = \{x \mid \text{Dom}(\Phi_x) \text{is finite}\}$;
- $\text{NOTTOT} = \{x \mid \Phi_x \text{is not total}\}$, etc.

For instance, to prove that EMPTY is not r.e. we need to show that there is $m \in$ EMPTY, $m' \notin$ EMPTY such that $\Phi_m \subseteq \Phi_{m'}$. Choose Φ_m to be the empty function and $\Phi_{m'}$ to be any function with domain $\{0\}$. Both functions are computable and $\Phi_m \subseteq \Phi_{m'}$. Therefore, EMPTY is not r.e.