Rice's Theorem

Every property of programs which concerns the I/O behaviour is undecidable.

```
" P is terminating on every input "
" P has some fixed m ∈ N as an output "
" P computes a function f "
...

" the length of program P is ≤ 10 "
```

What is a behavioural property of a program?

```
A ⊆ IN
↑
set of programs
(program property)
```

```
T = \{ m | P_m is terminating on every input \}
= \{ m | P_m is total \}
```

```
\text{ONE} = \{ m | \text{P_m is a sound implementation of } \Pi \}
= \{ m | \text{P_m is } \Pi \}
```

A ⊆ IN (program property) is a behavioural property if for all programs m ∈ IN
the fact that m ∈ A or m ∉ A

only depends on P_m
Def. (saturated / extensional set): $A \in \mathbb{N}$ is saturated (extensional) if for all $m, n \in \mathbb{N}$

if $m \in A$ and $\varphi_m = \varphi_n$ then $m \in A$

$\supseteq$

A saturated if $A = \{ m \mid \varphi_m \text{ satisfies a property of functions} \}$

= $\{ m \mid \varphi_m \in \mathcal{A} \}$

where $\mathcal{A} = \{ \varphi \}$ set of all functions

Examples

$\times \ \mathcal{T} = \{ m \mid \varphi_m \text{ is terminating on every input} \}$

= $\{ m \mid \varphi_m \text{ is total} \}$

= $\{ m \mid \varphi_m \in \mathcal{C} \}$

$\mathcal{C} = \{ f \in \mathcal{F} \mid f \text{ total} \}$

$\times \ \text{ONE} = \{ m \mid \varphi_m \text{ is a sound implementation of 1} \}$

= $\{ m \mid \varphi_m = 1 \}$

= $\{ m \mid \varphi_m \in \{ 1 \} \}$

$\times \ \text{LEN}^{10} = \{ m \mid \varphi_m \text{ has length } \leq 10 \}$

$m, n \in \text{LEN}^{10}$

and $\varphi_m = \varphi_n$

$m \notin \text{LEN}^{10}$

Ex. $m = \varphi\left(\{1\}\right) \in \text{LEN}^{10}$

$\varphi_m = \varphi_n = 0$

$\uparrow$

constant

zero

$m = \varphi\left(\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}\right) > 11 \notin \text{LEN}^{10}$
\[ K = \{ m \mid \varphi_m(m) \downarrow \} = \{ m \mid \varphi_m \in \mathcal{G}_1 \} \]

It seems that \( K \) is not saturated.

Formally, I should find \( m, m \in \mathbb{N} \)

- \( m \in K \quad \varphi_m(m) \downarrow \)
- \( m \notin K \quad \varphi_m(m) \uparrow \)

and \( \varphi_m = \varphi_m \)

If we were able to show that there is program \( m \in \mathbb{N} \) s.t.

\[ \varphi_m(x) = \begin{cases} 1 & \text{if } x = m \\ \uparrow & \text{otherwise} \end{cases} \]

we can conclude:

1. \( m \in K \quad \varphi_m(m) \downarrow \)

2. For a computable function there are infinitely many programs hema hure is \( m \neq m \) s.t. \( \varphi_m = \varphi_m \)

3. \( m \notin K \quad \varphi_m(m) \uparrow \quad \varphi_m(m) \uparrow \Rightarrow \varphi_m = \varphi_m \quad m \neq m \)

\( K \) is not saturated!

What about \( (\ast) \)?

\[ x \xrightarrow{\varphi} \text{P} \quad \begin{cases} \text{if } x = \text{P} \text{ then } 1 \\ \text{if } x \neq \text{P} \text{ then } \uparrow \end{cases} \]

\[ \text{def } P(x) : \quad \begin{cases} \text{if } x = \text{"def } P(x) : \text{..."} \\ \quad \text{if } x \neq \text{P} \text{ then } \uparrow \end{cases} \]
Rice’s Theorem:

Let \( A \subseteq \mathbb{N} \) if \( A \) is nonempty, \( A \neq \emptyset, A \neq \mathbb{N} \)

then \( A \) is not recursive.

Proof:

we show \( K \leq_m A \) (since \( K \) is not recursive, \( \neg m \), \( A \) not recursive)

Let \( e_0 \in \mathbb{N} \) be s.t. \( \varphi_{e_0}(x) \uparrow \forall x \) (program for the function always undefined)

1. Assume \( e_0 \notin A \)

   let \( e_1 \in A \) (it exists since \( A \neq \emptyset \))

   define

   \[
   g(x, y) = \begin{cases} 
   \varphi_{e_1}(y) & \text{if } x \in K \\
   \varphi_{e_0}(y) & \text{if } x \in \overline{K} 
   \end{cases}
   \]

   \[
   = \begin{cases} 
   \varphi_{e_1}(y) & \text{if } x \in K \quad [\varphi_x(x) \downarrow] \\
   \uparrow & \text{if } x \in \overline{K} \quad [\varphi_x(x) \uparrow]
   \end{cases}
   \]

   \[
   = \varphi_{e_1}(y) \land \varphi_x(x)
   \]

   computable!
By smm theorem there is \( S : \mathbb{N} \to \mathbb{N} \) total and computable s.t. \( \forall x, y \)

\[
\phi_{S(x)}(y) = g(x, y) = \begin{cases} 
\phi_{e_1}(y) & \text{if } x \in K \\
\phi_{e_0}(y) & \text{if } x \notin K 
\end{cases}
\]

\( S \) is the reduction function for \( K \subseteq \mu A \)

\* \( x \in K \implies S(x) \in A \)

if \( x \in K \) then \( \phi_{S(x)}(y) = g(x, y) = \phi_{e_1}(y) \quad \forall y \)

i.e. \( \phi_{S(x)} = \phi_{e_1} \). Since \( e_1 \in A \) and \( A \) saturated \( \iff S(x) \in A \)

\* \( x \notin K \implies S(x) \notin A \)

if \( x \notin K \) then \( \phi_{S(x)}(y) = g(x, y) = \phi_{e_0}(y) \quad \forall y \)

i.e. \( \phi_{S(x)} = \phi_{e_0} \). Since \( e_0 \notin A \) and \( A \) saturated \( \iff S(x) \notin A \)

Hence \( S \) is the reduction function for \( K \subseteq \mu A \) and since \( K \) not recursive, we deduce \( A \) not recursive.

2) if instead \( e_0 \notin A \)

\( e_0 \notin \bar{A} \)

\( \bar{A} \) saturated (since \( A \) is saturated)

\( \bar{A} \neq \emptyset \) (since \( A \neq \emptyset \))

\( \bar{A} \neq \emptyset \) (since \( A \neq \emptyset \))

\( \therefore \) by (1) applied to \( \bar{A} \) we deduce \( \bar{A} \) not recursive

\( \therefore A \) not recursive (since \( A \) recursive \( \Rightarrow \bar{A} \) recursive)
* Output problem \( B_m = \{ x \mid m \in E_x \} \)

we observed \( k \leq_m B_m \)

- \( B_m \) saturated,
  - in fact
  
  \( B_m = \{ x \mid \varphi_x \in B_m \} \)
  
  \( B_m = \{ f \mid m \in \text{odd}(f) \} \)

- \( B_m \neq \emptyset \)
  
  e.g.
  
  let \( e_1 \in \mathbb{N} \) be s.t.
  
  \( \varphi_{e_1}(y) = y \quad \forall y \quad \Rightarrow \quad m \in E_{e_1} = \mathbb{N} \)

  \( \Rightarrow e_1 \in B_m \neq \emptyset \)

- \( B_m \neq \mathbb{N} \)
  
  e.g.
  
  let \( e_2 \in \mathbb{N} \) s.t.
  
  \( \varphi_{e_2}(y) = m \quad (\neq m) \quad \forall y \)

  \( e_2 \in B_m \quad (\text{since } m \notin E_{e_2} = \{m\} \) \)

  \( \Rightarrow \) By Rice's Theorem \( B_m \) is not recursive.

**Example:**

\( I = \{ x \in \mathbb{N} \mid \varphi_x \text{ has infinitely many possible outputs} \} \)

\( = \{ x \in \mathbb{N} \mid E_x \text{ is infinite} \} \)

* saturated

\( I = \{ x \mid \varphi_x \in Y \} \)

with \( Y = \{ f \mid \text{odd}(f) \text{ infinite} \} \)

* \( I \neq \emptyset \)
  
  if \( e_1 \) is as in previous exercise \( \Rightarrow E_{e_1} = \mathbb{N} \text{ infinite} \quad \Rightarrow \quad e_1 \in I \)

* \( I \neq \mathbb{N} \)
  
  if \( e_2 \) is as before \( \Rightarrow E_{e_2} = \{m\} \quad \Rightarrow \quad e_2 \notin I \)

\( \Rightarrow \) \( I \) is not recursive, by Rice's Theorem.
Example

\[ A = \{ x \mid x \in W_x \cap E_x \} \]

saturated?

\[ A = \{ x \mid p_x \in A \} \]

\[ A = \{ f \mid ? \in \text{dom}(f) \cap \text{cod}(f) \} \]

we do not know what to put here

probably not saturated

we do not use Rice

We \( K \leq_m A \), i.e. that there is a total computable function \( s : \mathbb{N} \rightarrow \mathbb{N} \)

s.t.

\[ x \in K \iff s(x) \in A \]

\[ s(x) \in W_{s(x)} \quad \ldots \quad p_{s(x)}(s(x)) \downarrow \]

and

\[ s(x) \in E_{s(x)} \quad \ldots \quad p_{s(x)}(y) = s(x) \quad \text{for some } y \]

we define

\[ g(x, y) = \begin{cases} y & \text{if } x \in K \\ \uparrow & \text{otherwise} \end{cases} \]

\[ = y \cdot \downarrow \left( p_x(x) \right) \]

\[ = y \cdot \downarrow \left( p_y(x, y) \right) \quad \text{computable} \]

By smm theorem there is \( s : \mathbb{N} \rightarrow \mathbb{N} \) total computable s.t.

\[ p_{s(x)}(y) = g(x, y) = \begin{cases} y & \text{if } x \in K \\ \uparrow & \text{otherwise} \end{cases} \quad \forall x, y \]

is the reduction function

\[ \rightarrow \text{ if } x \in K \text{ then } p_{s(x)}(y) = g(x, y) = y \quad \forall y \]

Hence

\[ s(x) \in W_{s(x)} \cap E_{s(x)} = \mathbb{N} \quad \text{Thus } s(x) \in A \]

\[ \mathbb{N} \cap \mathbb{N} \]
\[ \rightarrow \text{ if } x \notin K \text{ then } \psi_{s(x)}(y) = g(x,y) \uparrow \forall y \]

Hence

\[ S(x) \notin W_{s(x)} \cap E_{s(x)} = \emptyset \]

Thus

\[ s(x) \notin A \]

Thus \( K \leq_m A \), and, since \( K \) not recursive, also \( A \) is not recursive.