Lessons touched by this meeting according to schedule:

- 16. 02/12/2024
  - Rice's theorem [§6.1.7, §6.1.8]
- 17. 03/12/2024
  - Recursively enumerable sets [§7.2.1, §7.2.2, §7.2.4, §7.2.5, §7.2.6, §7.2.13]

Let's start from the basics:

Definition

A subset $A \subseteq \mathbb{N}$ is saturated (or extensional) if $\forall\, m, n \in \mathbb{N}$

$$if\ m \in A\ and\ \phi_m = \phi_n\ then\ n \in A$$

(in words:

- given two programs, if the first program is in the set of programs satisfying the property and two programs are computing the same thing, then also the second program satisfies the property
- this means that if one program with a certain property is in the set, all programs computing the same function must also be in the set)

The property does not depend on the program but on the function it computes. A saturated set contains all the indices (which are infinite) of programs that compute functions with a particular common characteristic.

$\Updownarrow$

A saturated if $A = \{n \mid \phi_n \text{ satisfies a property of functions}\} = \{n \mid \phi_n \in A\}$

where $A \subseteq F$

property of functions — set of all functions

Examples

- $T = \{n \mid P_n \text{ is terminating on every input}\}$

There are some examples which require the use of a universal function:

Examples :

* $T = \{m \mid P_m \text{ always terminate (on every input)}\}$ SATURATED
  $= \{m \mid \varphi_m \text{ is total}\}$
  $= \{m \mid \varphi_m \in \mathcal{T}\}$ where $\mathcal{T} = \{f \mid f \text{ is total}\}$

* ONE $= \{m \mid P_m \text{ computes } \mathbb{1}\}$
  $= \{m \mid \varphi_m = \mathbb{1}\}$
  $= \{m \mid \varphi_m \in \{\mathbb{1}\}\}$ SATURATED

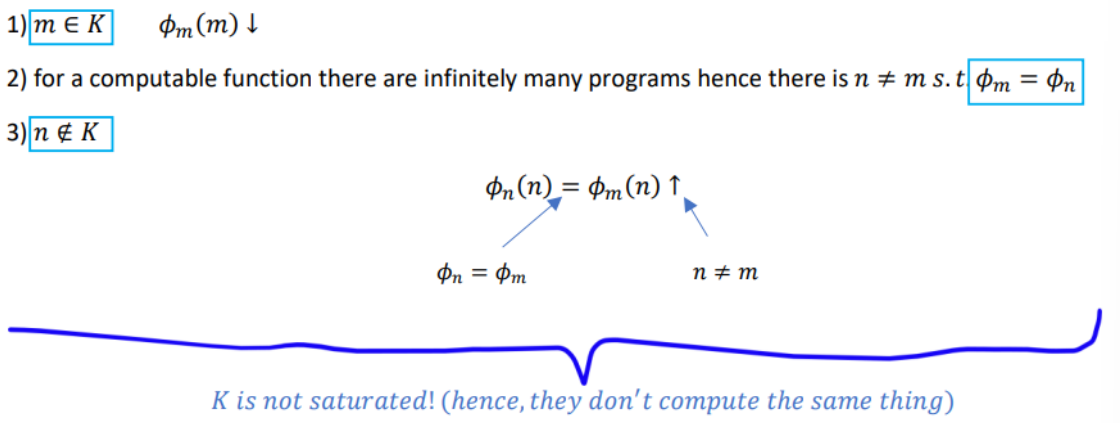
* LEN10 $= \{m \mid P_m \text{ length of } P_m \leq 10\}$ NOT SATURATED

  $m \in$ LEN10

  and $\varphi_m = \varphi_m$

  $m \notin$ LEN10

But there is a set, which is the halting set K, in which there is a “non-trivial” property, in which “infinitely many programs” are not able to compute on the same index:



Hence, we need something able to express such thing:

Definition (Rice’s theorem)

Let A be a set and $A \neq \emptyset, A \neq \mathbb{N}, A \subseteq \mathbb{N}$. If A is saturated, then A is not recursive.

In simpler terms:

- the saturation property implies that A contains all the indices of programs that compute functions with a common characteristic
  - this property holds extensionally, meaning it solely depends on the elements within the se without consideration for their internal representations.
- the significance of this in proving non-recursiveness lies in the inherent uncertainty: we cannot definitively determine whether a program possesses a specific property precisely

> In computability theory, **Rice's theorem** states that all non-trivial semantic properties of programs are undecidable. A *semantic* property is one about the program's behavior (for instance, "does the program terminate for all inputs?"), unlike a syntactic property (for instance, "does the program contain an if-then-else statement?"). A *non-trivial* property is one which is neither true for every program, nor false for every program.
>
> The theorem generalizes the undecidability of the halting problem. It has far-reaching implications on the feasibility of static analysis of programs. It implies that it is impossible, for example, to implement a tool that checks whether a given program is correct, or even executes without error (it is possible to implement a tool that always overestimates or always underestimates e.g. the correctness of a program, so in practice one has to decide what is less of a problem).

The proof goes on by reduction and requires the usage of a program index, in order to properly handle all cases of the function; this requires studying properties on "all" set, so to use both reduction and smn-theorem combined.

Let's see this by using an exercise:

Exercise 8.9. Study the recursiveness of the set $A = \{x \in \mathbb{N} \mid \varphi_x(y) = x * y \text{ per some } y\}$, that is to say if A e $\bar{A}$are recursive/recursively enumerable.

Solution: The set A is r.e. In fact the semi-characteristic function

$$_A(x) = \mu w.S(x, (w)_1, x * (w)_1, (w)_2)$$

is computable.

It is not is recursive, since $K \leqslant_m A$. In fact, consider the function

$$g(x, y) = \begin{cases} 0 & x \in K \\ \uparrow & \text{otherwise} \end{cases} = \mathbf{0}(sc_K(x))$$

It is computable and thus, by the smn theorem, we deduce that there is a total computable function $s : \mathbf{N} \to \mathbf{N}$ such that, for each $x, y \in \mathbb{N}$,

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

Then s is a reduction function for $K \leqslant_m A$. In fact

- if $x \in K$ then $\varphi_{s(x)}(y) = g(x, y) = 0$ for each $y \in \mathbb{N}$. In particular $\varphi_{s(x)}(0) = 0 = s(x) * 0$. Thus $s(x) \in A$.
- if $x \notin K$ then $\varphi_{s(x)}(y) = g(x, y) \uparrow$ for each $y \in \mathbb{N}$. Therefore surely there is no y such that $\varphi_{s(x)}(y) = x * y$. Thus $s(x) \notin A$.

Finally, since A r.e. and non-recursive, we conclude that $\bar{A}$ is not r.e. and thus not recursive. □

A set $A \subseteq N$ is recursively enumerable (called from now on "r.e.") if the semi-characteristic function $sc_A: \mathbb{N} \to \mathbb{N}$:

$$sc_A(x) = \begin{cases} 1, & x \in A \\ \uparrow, & otherwise \end{cases} \quad is\ computable$$

- Enumerable since there is surjective function total and computable $f: \mathbb{N} \to \mathbb{N}$ s.t. $cod(f) = A$
- Recursive since the enumeration can be done via a computable function

A property/predicate $Q(\vec{x}) \subseteq \mathbb{N}^k$ is semi-decidable if

$$sc_Q: \mathbb{N}^k \to \mathbb{N}$$

$$sc_Q(\vec{x}) = \begin{cases} 1, & if\ Q(\vec{x}) \\ \uparrow, & otherwise \end{cases} \quad computable$$

Keep in mind that a recursive set is said to be decidable, a r.e. set is said to be semidecidable.

The following is fundamental:

A language is recursively enumerable (r.e.) if it is the set of strings accepted by some TM. A language is recursive if it is the set of strings accepted by some TM that halts on every input. For example, any regular language is recursive.

Now, see how we use Rice's theorem:

Exercise 8.25. Let $\varnothing$ be the always undefined function. Study the recursiveness of the set $A = \{x \mid \varphi_x = \varnothing\}$, i.e., establish if A and $\bar{A}$ are recursive/recursive enumerable.

Solution: The set A is non-recursive, by Rice's theorem, since it is saturated, not empty (the always undefined function is computable) and different from $\mathbb{N}$.
In addition $\bar{A}$ is r.e., since

$$sc_{\bar{A}}(x) = \mathbf{1}(\mu w.H(x,(w)_1,(w)_2)$$

Thus A not r.e. □

Two notable functions here (useful for Rice-Shapiro):

The identity function is defined as:

```
id(x) = x for all x ∈ N
```

This function plays a crucial role in many proofs, particularly when working with saturated sets. It has several important properties:

1. It is total and computable
2. It has infinitely many indices (programs that compute it)
3. It is often used as a "reference point" when proving properties about sets of computable functions

The always undefined function, often denoted as ∅ or H, is defined as:

```
∅(x)↑ for all x ∈ N
```

This function has several crucial properties:

1. It is partial computable
2. Its domain is empty: dom(∅) = ∅
3. It is a subfunction of every function
4. It has infinitely many indices

Let's comment the following ones, which are important:

Exercise 8.29. Let $A = \{x \in \mathbb{N} : W_x \cap E_x \neq \varnothing\}$. Study the recursiveness of A, i.e., say if A and $\bar{A}$ are recursive/recursively enumerable.

Solution: The set A is saturated, since $A = \{x : \varphi_x \in \mathcal{A}\}$, where $\mathcal{A} = \{f \in \mathcal{C} : dom(f) \cap cod(f) \neq \varnothing\}$. It is not empty (since $\mathbf{1} \in \mathcal{A}$) and it is not the entire $\mathbb{N}$ (since $\varnothing \notin \mathcal{A}$), thus by Rice's theorem A is not recursive. Furthermore,A is r.e. since

$$\begin{aligned} sc_A(x) &= \mathbf{1}(\mu(y,z,t).H(x,y,t) \wedge S(x,z,y,t)) \\ &= \mathbf{1}(\mu w.H(x,(w)_1,(w)_3) \wedge S(x,(w)_2,(w)_1,(w)_3) \end{aligned}$$

Therefore $\bar{A}$ is not is r.e. □

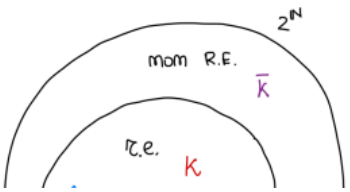
Universal quantification:

$$Q(t,\vec{x}) \subseteq \mathbb{N}^{k+1} \quad \text{decidable}$$

$$P(\vec{x}) \equiv \exists t.\, Q(t,\vec{x}) \quad \text{semi-decidable}$$

So, in words: if a decidable predicate is universally quantified existentially, it can become semi-decidable.

Definition (projection theorem) – closure by existential quantification



Let $P(x,\vec{y}) \subseteq \mathbb{N}^{k+1}$ semi-decidable

Then $R(\vec{y}) \equiv \exists x.\, P(x,\vec{y})$ is semi-decidable

Exercise 8.24. Study the recursiveness of the set

$$B = \{x \mid k \cdot (x+1) \in W_x \cap E_x \text{for each } k \in \mathbb{N}\},$$

i.e., establish if B and $\bar{B}$ are recursive/recursive enumerable.

Solution: The set A is not r.e., since $\bar{K} \leqslant_m A$. We prove it by considering

$$g(x,y) = \begin{cases} y & \neg H(x,x,y) \\ \uparrow & \text{otherwise} \end{cases}$$

This is computable and, by using the smn theorem, one can obtain the reduction function.

Also $\bar{A}$ is not r.e., since $\bar{K} \leqslant_m \bar{A}$. The reduction function can be obtained by considering

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

□

Consider these exercises as examples:

c. Given a function $f : \mathbb{N} \to \mathbb{N}$, consider the predicate $Q_f(x,y) \subseteq \mathbb{N}^2$ defined by $Q_f(x,y) \equiv$ "$f(x) = y$", where it is intended that if $f(x) \uparrow$ then $Q_f(x,y)$ is false. Show that f is computable if and only if Q_f is semi-decidable.

3. Assume that $f : \mathbb{N} \to \mathbb{N}$ is computable. If e is an index for f, i.e., $f = \varphi_e$ then $Q_f(x, y) \equiv \exists t.\, S(e, x, y, t)$. Since S is decidable, by the Structure Theorem, Q_f is semidecidable.

   Conversely, assume that $Q_f(x, y)$ is semidecidable, i.e., $sc_{Q_f} : \mathbb{N}^2 \to \mathbb{N}$ is computable. Then to compute $f(x)$ we just need to search y such that $Q_f(x, y)$ holds, i.e., $sc_{Q_f}(x, y) = 1$. Since sc_{Q_f} is partial we need to use the step predicates. Let e be an index for sc_{Q_f}, i.e., $sc_{Q_f} = \varphi_e^{(2)}$. Then

Exercise (2022-06-17)

c. Show that if predicate $Q(\vec{x}, y) \subseteq \mathbb{N}^{k+1}$ is semi-decidable then also $P(\vec{x}) = \exists y.Q(\vec{x}, y)$ is semi-decidable (do not assume structure and projection theorems). Does the converse hold, i.e., is it the case that if $P(\vec{x}) = \exists y.Q(\vec{x}, y)$ is semi-decidable then $Q(\vec{x}, y)$ is semi-decidable? Provide a proof or a counterexample.

Solution

3. Let $Q(\vec{x}, y) \subseteq \mathbb{N}^{k+1}$ be semi-decidable. Then the semi-characteristic function $sc_Q : \mathbb{N}^{k+1} \to \mathbb{N}$ is computable. Let $e \in \mathbb{N}$ be such that $sc_Q = \varphi_e^{(k+1)}$.

   Then $Q(\vec{x}, y)$ holds iff $\varphi_e^{(k+1)}(\vec{x}, y) = 1$ iff $\varphi_e^{(k+1)}(\vec{x}, y) \downarrow$ iff $\exists t.H^{(k+1)}(e, (\vec{x}, y), t)$.
   Therefore $Q(\vec{x}, y) \equiv \exists t.\, H^{(k+1)}(e, (\vec{x}, y), t)$ and thus

$$P(\vec{x}) \equiv \exists y.\, Q(\vec{x}, y) \equiv \exists y.\exists t.\, H^{(k+1)}(e, (\vec{x}, y), t) \equiv \exists w.\, H^{(k+1)}(e, (\vec{x}, (w)_1), (w)_2)$$

   Therefore $sc_P(\vec{x}) = \mathbf{1}(\mu w.\ |\chi_{H^{(k+1)}}(e, (\vec{x}, (w)_1)), (w)_2) - 1|)$ is computable, and thus $P(\vec{x})$ is semi-decidable.

   The converse implication does not hold. For instance, consider the predicate $Q(x, y) \equiv$ "$\phi_y(x) \uparrow$". Then $P(x) \equiv \exists y.Q(x, y) \equiv \exists y.\phi_y(x) \uparrow$ is always true, hence decidable. In fact, if e_0 is an index for the always undefined function, for $y = e_0$ clearly $Q(x, y)$ for every $x \in \mathbb{N}$. Instead $Q(x, y) = \phi_y(x) \uparrow$ is not semi-decidable (it is negation of the halting predicate, which is semi-decidable but not decidable).

Some other exercises:

Exercise 6.9. Is there non-computable function $f : \mathbb{N} \to \mathbb{N}$ such that $dom(f) \cap img(f)$ is empty? Justify your answer (providing an example of such f, if it exists, or proving that cannot exist).

Solution: Consider the function

$$f(x) = \begin{cases} 2 * \chi_K(\lfloor x/2 \rfloor) & \text{if } x \text{ odd} \\ \uparrow & \text{otherwise} \end{cases}$$

We have that $dom(f)$ is the set of odd numbers, $cod(f) = \{0, 2\}$, then $dom(f) \cap cod(f) = \varnothing$. Also, f is not computable. If it were then also $\chi_K(z) = f(2z+1)/2$ would be computable, while we know that K is not recursive, i.e., χ_K is not is computable. □