(7/25)

6 Uncomputability

6.1 Diagonal method

Note
$comp$ is a not necessarily rec func.
Let's extend it to a total function.
That is,

$$comp^{+}(p,x) = \begin{cases} y & \text{if } comp(p,x) \text{ is defined and its value is y} \\ 0 & \text{if } comp(p,x) \text{ is undefined} \end{cases}$$

Thm6.1
$comp^{+} : \mathbb{N}^2 \to \mathbb{N}$ is not recursive.
Proof
By the diagonal method.
Consider the function

$$diag(x) := comp^{+}(x, < x >) + 1$$

By contradiction,
Assume $comp^{+}$ is a rec func.
By def, $comp^{+}$ is total,too.

Then

$$diag : \mathbb{N} \to \mathbb{N}$$

is also a total rec func.

Since it is recursive, we can take a code N of diag.
Then we have

$$diag(x) = comp(N, < x >) = comp^{+}(N, < x >) \tag{1}$$

Now,

$$\begin{aligned} diag(N) &\xlongequal[\text{def of } diag]{} comp^{+}(N, < N >) + 1 \\ &\xlongequal[\text{by (1)}]{} diag(N) + 1 \\ &\longrightarrow \text{comtradiction!} \end{aligned}$$

Cor6.2

$halt(p, x) \iff$

- p is a code of a recursive function of arity n, for some n.
- $x =< \vec{y} >$, $\vec{y}$ is of length n.
- The function p is defined for input $\vec{y}$.

Then, $halt(p, x)$ is not recursive.

Proof

If $halt$ is recursive , then

$$comp^{+}(p, x) = \begin{cases} comp(p, x) & \text{if } halt(p, x) \text{ is true.} \\ 0 & \text{if } halt(p, x) \text{ is false.} \end{cases}$$

is also recursive.

$\longrightarrow$contradiction.

Other uncomputable (not-recursive) predicates:

- $total(p)$ Does p code a total function?
- $equal_f(p)$ Does p code the same function as given f?

6.2 Some Results on (Un)computability

- Parameter Theorem (A.k.a s-m-n Theorem)
- Recursion Theorem
- Rice's Theorem

6.3 Recursively Enumerable Predicates(帰納的枚挙可能)

Def6.3

A 1-ary predicates $P \subseteq \mathbb{N}$ is recursively enumerable(RE)
$\overset{\text{def}}{\Longleftrightarrow}$There is a 2-ary predicate Q s.t. $P(x) \Leftrightarrow \exists y Q(x, y)$

Thm6.4

The following are equivalent.

1. P is RE

2. P is semi-decidable.(半決定可能)
   That is,
   There is a recursive function $f : \mathbb{N} \dashrightarrow \mathbb{N}$
   s.t. $f(x) = \begin{cases} 0 & \text{(if } P(x) \text{ is true)} \\ \perp & \text{(if } P(x) \text{ is false)} \end{cases}$

Thm6.5(Negation Theorem)

The following are equivalent.

1. P is decidable.

2. P and $\neg P$ are both RE.

Proof

[1⇒2].

$$\begin{array}{ccc} P \text{ is rec.} & \Longrightarrow & \neg P \text{ is rec.} \\ \Downarrow & & \Downarrow \\ P \text{ is RE.} & & \neg P \text{ is RE.} \end{array}$$

[2⇒1(Idea)].

$$\xrightarrow{\ x\ } \boxed{P?} \to \text{Yes}/ \perp$$
$$\xrightarrow{\ x\ } \boxed{\neg P?} \to \text{Yes}/ \perp$$

7 Gödel's imcompleteness

$$\underset{\text{LK"derivable"syntactic}}{\Vdash A} \quad \underset{complete}{\overset{sound}{\rightleftarrows}} \quad \underset{\text{"valid"semantical}}{\models A}$$

Φ: a set of formulas ("axioms")

$$\Phi \Vdash A \quad \Longrightarrow \quad \Phi \models A$$

$$\text{LK+}\Phi\text{"derivable"syntactic} \quad \underset{\substack{string\\completeness}}{\Longleftarrow} \quad \begin{array}{l}\text{if } I, J \text{ satisfy}\\ [\![B]\!] = t \text{ for } \forall B \in \Phi\\ \text{Then } [\![A]\!] = \mathrm{tt}\end{array}$$

Now, fix our symbols :

$$FnSymb = \{0, s, +, \cdot\}$$
$$PredSymb = \{=, <\}$$

Then we're more interested in

$$\underbrace{\Phi}_{\uparrow\text{Is there } Phi \text{ that is complete?}} \quad \Vdash A \xrightarrow{sound} \underbrace{Is}_{\text{Standard Structure}} \quad \models A$$

Q : Is there Φ

$$\Phi \Vdash A \Longleftrightarrow Is \models A?$$

A : Yes.

$$\Phi = \{A | Is \models A\}$$

Q2 : Then, what's the problem?
A2 : Is this $LK + \Phi$ "mathematical"/"syntactic" ?
whether $A \in \Phi$ or not is not "easily" checkable?

<u>Thm(Gödel's incompleteness)</u>

1. If Φ is $\underbrace{\text{recursive axiomatized}}_{A \in \Phi \text{ or not is decidable.}}$ and complete
then $\Phi \Vdash A$ or not is decidable.
2. Is $\models A$ is not recursive(decidable).
3. Therefore; there is no rec axiomatized Φ
s.t.

$$\Phi \Vdash A \Longleftrightarrow Is \models A$$