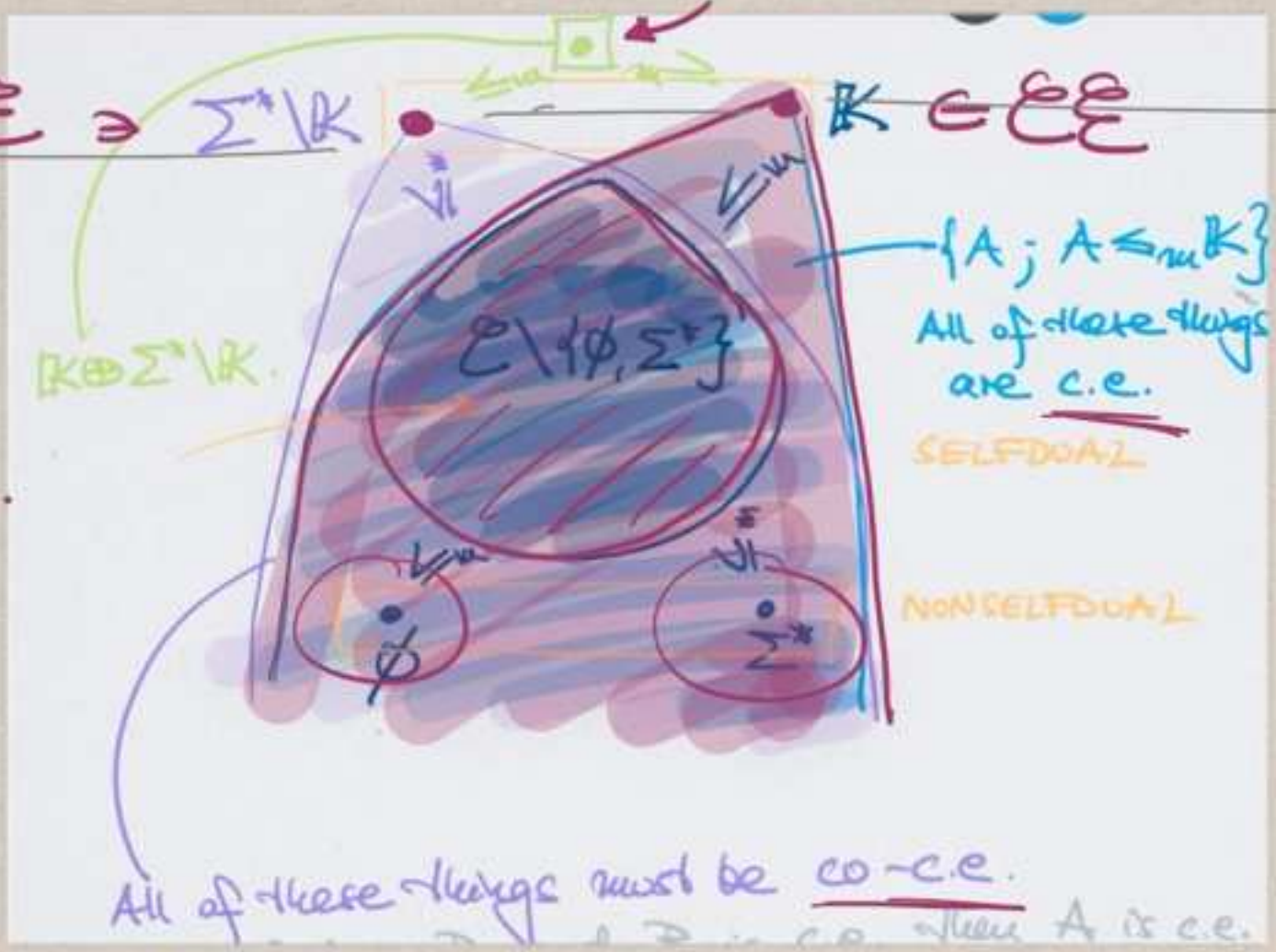
RECURSION THEORY

Lecture VII

21 DECEMBER 2021



DEGREES OF UNSOLVABILITY

$\equiv_m$

$\mathcal{E} = \{A \subseteq \Sigma^* ; A \text{ is computable}\}$

$\mathcal{E}\mathcal{E} = \{A \subseteq \Sigma^* ; A \text{ is c.e.}\}$

$\mathcal{E} = \mathcal{E}\mathcal{E} \cap$ co-$\mathcal{E}\mathcal{E}$

where co-$\mathcal{E}\mathcal{E} := \{A \subseteq \Sigma^* ; A \text{ is co-c.e.}\}$

Lectures in January

VIII	Wed 5 Jan	14-16
IX	Thu 6 Jan	16-18
X	Wed 12 Jan	14-16
XI	Thu 13 Jan	16-18

Reminders

$$w \longmapsto w_E$$
$$\sigma_0 \dots \sigma_{2n} \longmapsto \sigma_0\sigma_2\sigma_4 \dots \sigma_{2n}$$
$$\sigma_0 \dots \sigma_{2n+1} \longmapsto \sigma_0\sigma_2\sigma_4 \dots \sigma_{2n}$$

$$w \longmapsto w_O$$
$$\sigma_0 \dots \sigma_n \longmapsto \sigma_1\sigma_3 \dots \sigma_k$$

These are computable.

Their inverse would be:

$$(w_E, w_O) \longmapsto w$$

By definition, this cannot be a computable function since it is a function from $(\Sigma^*)^2$ to Σ^* and computability is only defined for functions from Σ^* to Σ^*.

Write $(w,v) \in (\Sigma^*)^2 \longmapsto w*v \in \Sigma^*$

s.t. $(w*v)_E = w$ and

$(w*v)_O = v$.

Then this operation is "computable" in the following (very strong) sense:

Fix w. Then the function

$$v \longmapsto w * v$$

is computable. Moreover, there is a computable function $c : \Sigma^* \to \Sigma^*$ s.t.

$$f_{c(w)}(v) = w * v$$

[There is a uniform way to write down a program that performs the operation $v \longmapsto w * v$.]

Remark An operation $(\Sigma^*)^2 \to \Sigma^*$ has been transformed into an operation that modifies programs.

This is known as CURRYING.

Theorem (s-m-n Theorem / Parameter Theorem)

If g is a partial computable function, then there is a total computable function h s.t. for all v, w

$$\varphi_{h(v)}(w) = g(v * w).$$

Proof. Fix v as input and describe what $h(v)$ is.

By the earlier discussion, we have a computable c s.t. $c(v)$ is a code for the function $w \longmapsto v * w$.

By the assumption, we have a code for g, say p.

Given programs $c(v)$ and p, write a program that, upon input w, produces $\varphi_{c(v)}(w)$ and feeds this into φ_p.

By the principle of concatenation, we

can computably produce a program $h(v)$ doing precisely that:

$$f_{h(v)}(w) = f_p(f_{c(v)}(w))$$
$$= f_p(v * w)$$
$$= g(v * w).$$

q.e.d.

This is the transformation of a function in two variables

$$(v, w) \longmapsto g(v * w)$$

into a function in one variable, depending on a parameter.

→ CURRYING

History. It is called s-m-n Theorem since the function h in its first version was called s_n^m. People still use this name, but not the notation.

Haskell Brooks Curry

Born	September 12, 1900 Millis, Massachusetts, US
Died	September 1, 1982 (aged 81) State College, Pennsylvania, US
Nationality	American

CURRYING

is the following observation:

If X, Y, Z are sets, then there is a canonical isomorphism between

$$Z^{X \times Y}$$

and

$$(Z^Y)^X.$$

In other words:

$$f : X \times Y \longrightarrow Z$$

can be considered as

$$\hat{f} : X \longrightarrow Z^Y.$$

Corollary If A is c.e., then $A \leq_m K$.

Proof. We know that there is some f partial computable s.t. $A = \mathrm{dom}(f)$.

Define

$$g(v * w) := \begin{cases} \varepsilon & \text{if } v \in A \\ \uparrow & \text{if } v \notin A. \end{cases}$$

(partial)

Claim g is computable. Given $v \in \Sigma^*$

Step 1 $v := v_E$.

Step 2 Compute $f(v)$.

If $f(v)\uparrow$, then our algorithm does not produce output.

If $f(v)\downarrow$. Then output ε.

This produces precisely the output of g. Thus g is computable. [q.e.d. Claim]

IMPORTANT REMARK

In general, a case distinction

$$g(x) := \begin{cases} y & \text{if } x \in A \\ z & \text{if } x \notin A \end{cases}$$

only produces a computable function if A is computable. This particular case distinction only works because $z = \uparrow$ and that is the bad outcome of trying to check whether $x \in A$.

[Compare the proofs of the computability of range and domain check!]

$$g(v*w) := \begin{cases} \varepsilon & \text{if } v \in A \\ \uparrow & \text{if } v \notin A. \end{cases}$$

is computable.

Therefore, the s-m-n Theorem gives a total computable k s.t.

$$f_{k(v)}(w) = g(v*w).$$

We now show that h is a many-one-reduction witnessing

$$A \leq_m \mathbb{K}.$$

$$f_{h(w)}(w) = \begin{cases} \downarrow & \text{if } w \in A \\ \uparrow & \text{if } w \notin A \end{cases}$$

[Have to show
$w \in A \Rightarrow h(w) \in \mathbb{K}$
$w \notin A \Rightarrow h(w) \notin \mathbb{K}$.]

(i) Suppose $w \in A$. Then $f_{h(w)}$ is the constant function $w \longmapsto \varepsilon$.
Thus $f_{h(w)}(w)\downarrow$ for every w.
Thus $f_{h(w)}(h(w))\downarrow$, so $h(w) \in \mathbb{K}$.

(ii) Suppose $w \notin A$. Then $f_{h(w)}$ is the partial fn that is nowhere defined.
Thus $f_{h(w)}(w)\uparrow$ for every w.
So $f_{h(w)}(h(w))\uparrow$, so $h(w) \notin \mathbb{K}$.

q.e.d.

Corollary

$$\mathcal{EE} = \{A;\ A \leq_m K\}.$$

and

$$\text{co-}\mathcal{EE} = \{A;\ A \leq_m \Sigma^* \setminus K\}.$$

Definition Let $\mathcal{Z} \subseteq \mathcal{P}(\Sigma^*)$. We call $A \subseteq \Sigma^*$ $\mathcal{Z}$-hard if for all $B \in \mathcal{Z}$, we have $B \leq_m A$. We call A $\mathcal{Z}$-complete if A is $\mathcal{Z}$-hard and $A \in \mathcal{Z}$.

Remarks

1. This will only exist if $\mathcal{Z}$ is countable.
   [Since only countably many B can satisfy $B \leq_m A$.]
2. We already proved that if $A \neq \emptyset, \Sigma^*$, then A is $\mathcal{E}$-hard and if $A \in \mathcal{E} \setminus \{\emptyset, \Sigma^*\}$, then A is $\mathcal{E}$-complete.

3. You could define hardness and completeness for any other type of reductions.

4. For instance, in complexity theory

$P = \{A;$ A can be computed in polynomial time$\}$

$NP = \{A;$ a computation of being in A can be checked in polynomial time$\}$

OPEN | The famous $P = NP$ problem asks is $P = NP$?

There are many NP-complete problems known. So it could be possible to prove $P = NP$ by reducing one of the NP-complete problems to a problem known to be in P.

Majority belief: This should be impossible But at the moment, we do not know.

$1,000,000 Clay Prize.

In this terminology, the earlier Corollary states:

$$K \text{ is } \mathcal{EE}\text{-complete}$$

and

$$\Sigma^* \setminus K \text{ is co-}\mathcal{EE}\text{-complete.}$$

INDEX SETS & RICE'S THM

We know that c.e. sets are domains of partial computable functions.

If P is a program, f_P is partial computable, and therefore

$$W_P := \text{dom}(f_P)$$

is c.e.

Moreover, $\mathcal{EE} = \{ W_P ; P \in \Sigma^* \}$.

Using this notation, write

$$K_0 := \{ v * v ; v \in W_v \}$$

This set is c.e. by universality.

Reminder:

Universality was the computability of

$$w \longmapsto f_{w_E}(w_0)$$

So K_0 is just the domain of Kleene's computable functions.

By our earlier corollary, this implies

$$K_0 \leq_m K \quad \left.\right\} \; K \equiv_m K_0$$

Claim. $K \leq_m K_0$.

[Use $h(v) := v * v$ as reduction function (REDUPLICATION):

Suppose $v \in K \Longrightarrow f_v(v)\downarrow$

$\Longrightarrow v \in \mathrm{dom}(f_v)$

$\Longrightarrow v \in W_v$

$\Longrightarrow h(v) = v * v \in K_0$

Suppose $v \notin K \Longrightarrow f_v(v)\uparrow$

$\Longrightarrow v \notin \mathrm{dom}(f_v) = W_v$

$\Longrightarrow h(v) = v * v \notin K_0$.]

Definition A set $A \subseteq \Sigma^*$ is called an *index set* if there is a $\mathcal{Z} \subseteq \mathcal{CE}$ s.t.

$$A = \{P_i : W_P \in \mathcal{Z}\}.$$

That means: A is a set of indices of some property of c.e. sets.

An index set A is called *trivial* if $A = \emptyset, \Sigma^*$.

Examples

$Emp = \{P_i : W_P = \emptyset\}$

$Nonemp = \{P_i : W_P \neq \emptyset\}$

$Fin = \{P_i : W_P \text{ is finite}\}$

$Inf = \{P_i : W_P \text{ is infinite}\}$

$Tot = \{P_i : \varphi_P \text{ is total}\}$

$= \{P_i : W_P = \Sigma^*\}$

Non-Examples K and K_0

(This will be proved after Christmas; probably lecture IX.)

Theorem The set

$$\{P;\ P \text{ is a program s.t. } W_P = \mathrm{dom}(f_P) \neq \emptyset\}$$

is c.e.

Proof. Remember from the argument for RANGE CHECK:

$i \longmapsto W_i$ computable assignment of the i-th word in Σ^*

$\langle i,j\rangle : \mathbb{N}^2 \longrightarrow \mathbb{N}$ computable bijection

I use the same trick as in the argument for RANGE CHECK:

In STEP n of the algorithm

find $n = \langle i,j\rangle$

Compute $f_P(w_i)$ for j steps (*)

If this has halted, then halt and output 1

If not, move to $n+1$.

$\Longrightarrow$ Nonemp is c.e.

$\Longrightarrow$ Nonemp $\leq_m K$.

$[\Longrightarrow$ Emp $\leq_m \overline{K}]$

Theorem (Rice's Theorem)

Every computable index set is trivial.

[Remark. This means that every set in

$$\mathcal{E} \setminus \{\emptyset, \Sigma^*\}$$

is a non-example. I.e., the set of words of even length.]

Remark 2. Nonemp is a non-computable c.e. set.

Proof of Rice's Theorem

We are going to show a more specific claim:

Let A be a nontrivial index set. We know that $\emptyset$ is c.e., so fix some P s.t. $W_P = \emptyset$.

We have two cases:

Case 1. $P \in A$ Case 2 $P \notin A$.

Case 1 $P \in A$. By non-triviality, find $Q \notin A$.

Consider

$$g(v*w) := \begin{cases} f_Q(w) & \text{if } v \in K \\ \uparrow & \text{if } v \notin K \end{cases}$$

not allowed case distinction

By the same argument we used before, g is partial computable.

So, by s-m-n, we find total h s.t.

$$f_{h(v)}(w) = g(v*w).$$

Claim h witnesses $\Sigma^* \setminus K \leq_m A$.

$$f_{h(w)}(w) = \begin{cases} f_Q(w) & \text{if } v \in K \\ \uparrow & \text{if } v \notin K. \end{cases}$$

Subcase 1a $v \in K \Longrightarrow$

$$f_{h(w)} = f_Q$$

Since A was an index set

$$h(v) \in A \iff Q \in A$$

So $h(v) \notin A$.

Subcase 1b $v \notin K \Longrightarrow$

$f_{h(w)}$ is nowhere defined

$\longrightarrow W_{h(w)} = \emptyset = W_p$)

Since A was an index set

$$h(v) \in A \iff P \in A.$$

But we're in case 1, so $h(v) \in A$.

So $\Sigma^* \setminus K \leq_m A$.

Case 2 $P \notin A$.

In this case, pick (by non-triviality) $Q \in A$.

Define the same function

$$g(v*w) = \begin{cases} f_Q(w) & \text{if } v \in K \\ \uparrow & \text{if } v \notin K \end{cases}$$

The same argument now gives

$$K \leq_m A.$$

q.e.d.

Observation If A is (non-trivial) index set and

$$A = \{P ; W_P \in Z\},$$

then if $\emptyset \in Z \Rightarrow \Sigma^* \setminus K \leq_m A$

$\emptyset \notin Z \Rightarrow K \leq_m A$.

Corollary $\emptyset \notin \text{Nonemp} \Rightarrow K \leq_m \text{Nonemp}$.

We proved earlier $\text{Nonemp} \leq_m K$.

$\Rightarrow K_1 := \text{Nonemp} \equiv_m K$

↑ the index set version of K.