Recursion Theory — THIRD LECTURE

3 NOVEMBER 2021

$\mathcal{M} = (\underline{\Sigma}, \underline{\Phi}, \underline{u})$ MODEL OF COMPUTATION

$\left.\begin{array}{l} \underline{P} \in \Sigma^* \\ \underline{w} \in \Sigma^* \end{array}\right\} \longrightarrow$ $\mathcal{M}$-computation of P with input w

$$f_P : \Sigma^* \dashrightarrow \Sigma^*$$
$$v \longmapsto \begin{cases} o_{\mathcal{M}}(P,w) & \text{if that computation halts} \\ \uparrow & \text{o/w} \end{cases}$$

f computable if there is a P s.t. $f = f_P$.

ADDITIONAL PROPERTIES

COMPOSITIONALITY if f, g computable, then $f \circ g$ is

CASE DISTINCTION if $x, y \in \Sigma^* \cup \{\uparrow\}$, then

$$\longrightarrow \quad \underline{d_{xy}} : v \longmapsto \begin{cases} x & \text{if } v = \varepsilon \\ y & \text{if } v \neq \varepsilon \end{cases}$$

is computable.

IDENTITY id: $\Sigma^* \longrightarrow \Sigma^*$ is computable.

$A, B \subseteq \Sigma^*$ sets of words called "problems".

many-one reducibility

$A \leq_m B :\Longleftrightarrow$ there is a total computable function f s.t.

f.a. $w \in \Sigma^* (w \in A \longleftrightarrow f(w) \in B.)$

FROM NOW ON, LET M BE A MODEL OF COMPUTATION SATISFYING COMPOSITIONALITY, CASE DISTINCTION, & IDENTITY.
[AND LATER ADDITIONAL PROPERTIES].

Remark. Reflexivity of $\leq_m$ uses the property of identity.

Some properties of $\leq_m$:

(1) $A \leq_m B \Longleftrightarrow \Sigma^* \setminus A \leq_m \Sigma^* \setminus B$.
[Just from the fact that $\leq_m$ is defined by an equivalence.]

(2) $\emptyset, \Sigma^*$ are computable.
[A set $A \subseteq \Sigma^*$ is computable if χ_A is computable.

$$\chi_A(w) := \begin{cases} \sigma & \text{if } w \in A \\ \varepsilon & \text{if } w \notin A. \end{cases}$$

Thus $\chi_\emptyset$ is the constant function that always assigns the empty word ε.

Let $x = y = \varepsilon$, then $d_{xy} = d_{\varepsilon\varepsilon} = \chi_\emptyset$.

Remark This shows that every constant function $const_x : w \longmapsto x$ is computable:

$$const_x = d_{xx}.$$

Including the case $x = \uparrow$: $d_{\uparrow\uparrow}$ is the function that is nowhere defined.

χ_{Σ^*} is the constant function $const_\sigma$, so Σ^* is also computable.]

(3) $\emptyset \not\leq_m \Sigma^*$.

[If $\emptyset \leq_m \Sigma^*$, then there is $f : \Sigma^* \longrightarrow \Sigma^*$ s.t. f.a. w

ALWAYS FALSE $\boxed{w \in \emptyset} \iff \boxed{f(w) \in \Sigma^*}$ ALWAYS TRUE

CONTRADICTION!

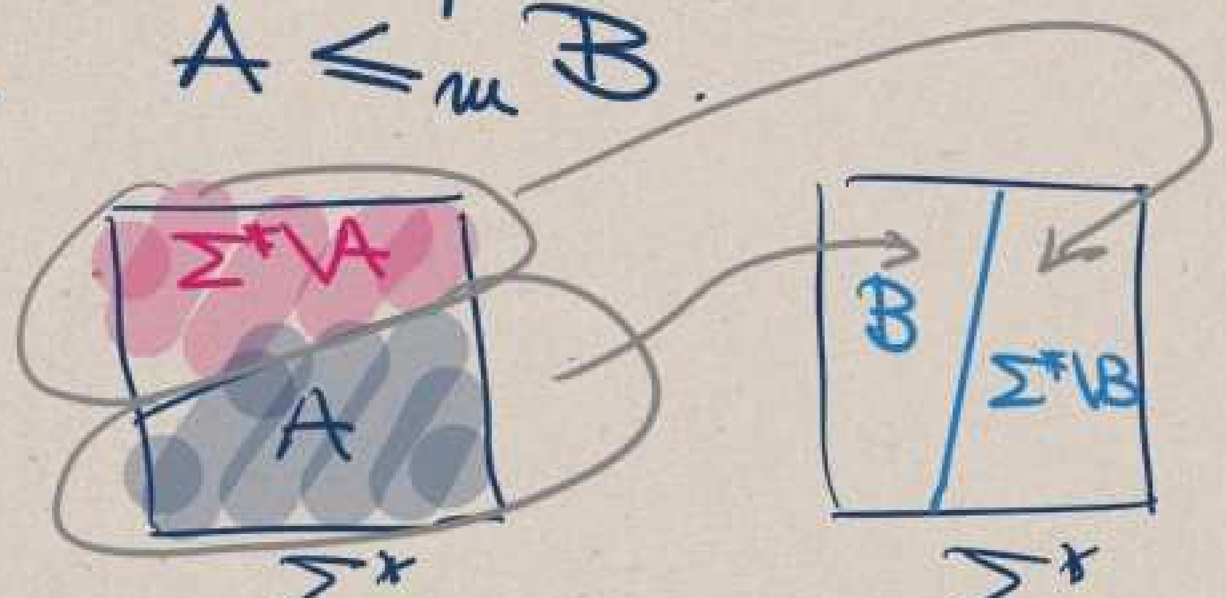
Remark. This is unrelated to computability: no function whatsoever can be a reduction of $\emptyset$ to Σ^*.

④ $\Sigma^* \not\leq_m \emptyset$

[From ③ & ①.]

⑤ If A is computable and $B \neq \emptyset, \Sigma^*$, then $A \leq_m B$.

Proof.



Since $B \neq \emptyset, \Sigma^*$, there are w and w' s.t. $w \in B$ and $w' \notin B$.

Since A is computable, χ_A is computable.

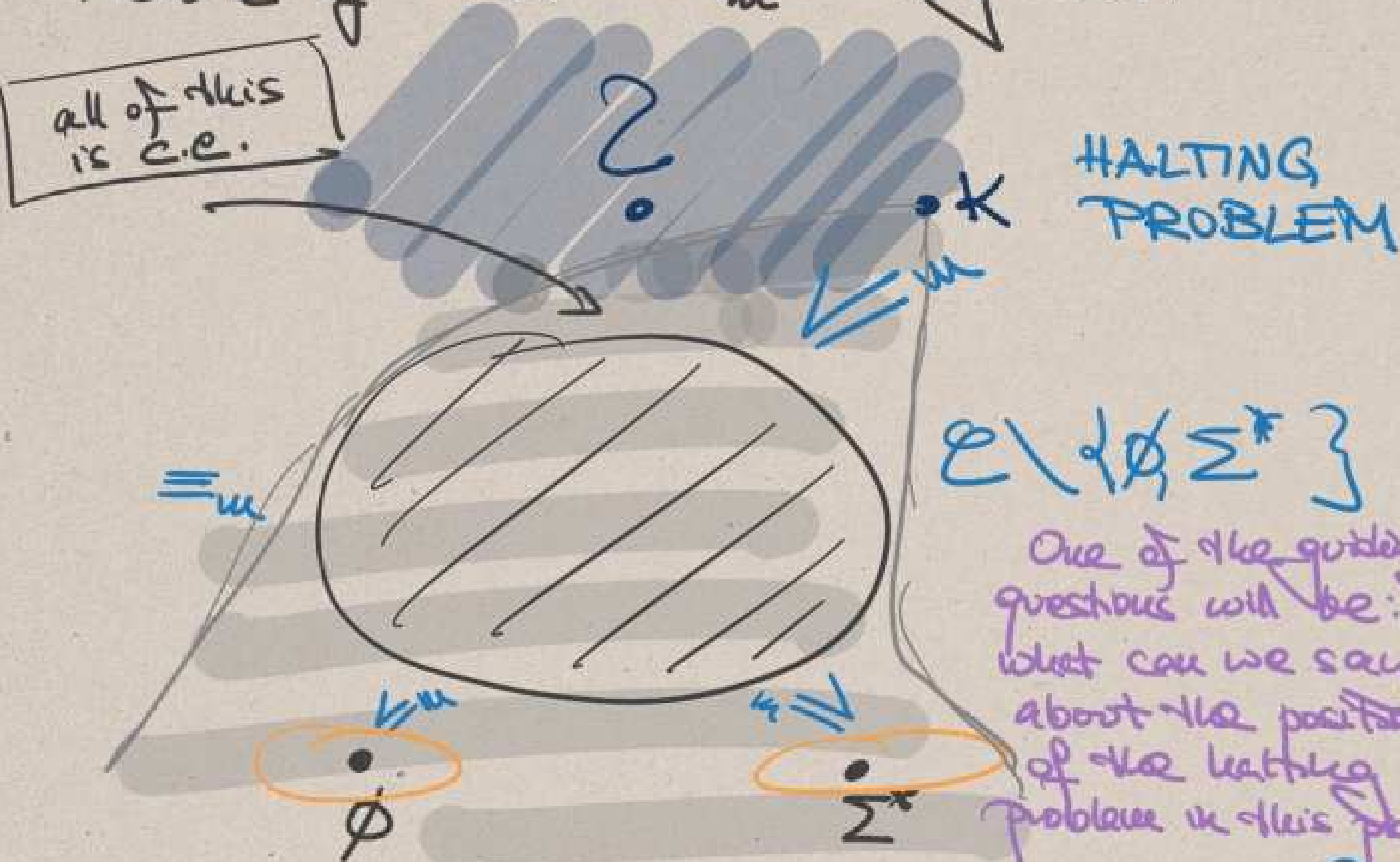
Consider

$$f := d_{w'w} \circ \chi_A : v \longmapsto \begin{cases} w' & \text{if } v \notin A \\ w & \text{if } v \in A \end{cases}$$

If $v \in \Sigma^*$, then $v \in A \longleftrightarrow f(v) \in B$.

q.e.d.

Picture of the $\equiv_m$-degrees:



One of the guiding questions will be: what can we say about the position of the halting problem in this picture?

$\mathcal{C} := \{A \subseteq \Sigma^* \,;\, A \text{ is computable}\}$

Argument that the picture is correct:

1. If $A \neq \emptyset$, then $A \not\leq_m \emptyset$. [Suppose $A \leq_m \emptyset$. By (4), $A \neq \Sigma^*$, so by (5), every computable set reduces to A, in particular
$$\Sigma^* \leq_m A \leq_m \emptyset.$$
By transitivity $\Sigma^* \leq_m \emptyset$. Contradiction to (4).]
2. If $A \neq \Sigma^*$, then $A \not\leq_m \Sigma^*$.
3. The rest of the picture is just (5) plus the fact that $A \leq_m B$ + B computable $\Longrightarrow$ A comput. [Lecture II]

We add more properties to our model of computation.

UNIVERSALITY

a.k.a. the SOFTWARE PRINCIPLE

The idea is that one machine can perform various tasks by changing the program.

If $w \in \Sigma^*$, say

$$w = \sigma_0 \sigma_1 \sigma_2 \dots \sigma_n ,$$

we can think of it as two words by separating into even and odd

w	w_E	w_O
$\sigma_0 \sigma_1 \dots \sigma_{2n}$	$\sigma_0 \sigma_2 \sigma_4 \dots \sigma_{2n}$	$\sigma_1 \sigma_3 \sigma_5 \dots \sigma_{2n-1}$
$\sigma_0 \sigma_1 \dots \sigma_{2n+1}$	$\sigma_0 \sigma_2 \sigma_4 \dots \sigma_{2n}$	$\sigma_1 \sigma_3 \sigma_5 \dots \sigma_{2n+1}$

$$w \longmapsto w_E : \Sigma^* \longrightarrow \Sigma^*$$
$$w \longmapsto w_O : \Sigma^* \longrightarrow \Sigma^*$$

should be computable. Furthermore, the function

$$u(w) := f_{w_E}(w_O)$$

is computable.

DUPLICATION

$$w = \sigma_0 \sigma_1 \sigma_2 \cdots \sigma_n$$

$$\downarrow$$

$$\sigma_0\sigma_0\sigma_1\sigma_1\sigma_2\sigma_2 \cdots \sigma_n\sigma_n$$

$$\shortparallel$$

$$r(w)$$

The function $r : \Sigma^* \longrightarrow \Sigma^*$ is computable.

REMEMBER

The halting problem K was

$$\{ w ; \underline{f_w(w)\downarrow} \}$$

$$f_w(w) = u \circ r(w)$$

$$K = \{ w ; u \circ r(w) \downarrow \}$$

$$= \mathrm{dom}(u \circ r)$$

<u>Summary</u> IF M satisfies UNIVERSALITY, DUPLICATION, COMPOSITIONALITY, then K is the domain of a partial computable function.

Definition A set A is called **COMPUTABLY ENUMERABLE** if there is a computable partial function f s.t.

$$A = \mathrm{dom}(f).$$

[Remark: We'll see later why "computably enumerable" is a good name for this.]

We'll now show a characterisation theorem for computably enumerable (c.e.) sets with four equivalent statements. Two of the directions will require additional properties of M.

Def. Let $A \subseteq \Sigma^*$. We call

$$\psi_A : w \longmapsto \begin{cases} \sigma & \text{if } w \in A \\ \uparrow & \text{if } w \notin A \end{cases}$$

the pseudocharacteristic function for A.

Note that

$$\psi_A = d_{\uparrow\varepsilon} \circ \chi_A,$$

so if χ_A is computable, then so is ψ_A.

Def. Let $A \subseteq \Sigma^*$. We call

$$\psi_A^* : u \longmapsto \begin{cases} u & \text{if } u \in A \\ \uparrow & \text{if } u \notin A \end{cases}$$

the strong pseudocharacteristic function for A.

THEOREM Let M be a model of computation satisfying all of the above properties plus RANGE CHECK and DOMAIN CHECK and $A \subseteq \Sigma^*$, then the following are equivalent:

(1) There is a computable f s.t. $A = \mathrm{dom}(f)$.

(2) ψ_A is computable

(3) ψ_A^* is computable

(4) There is a computable partial f s.t. $A = \mathrm{ran}(f)$.

[= COMPUTABLY ENUMERABLE]

The question about the name of the concept "c.e." is closely related to (4), but we would like a total computable function f s.t. $A = \mathrm{ran}(f)$.

Extra properties

DOMAIN CHECK

Let f, g be computable. Then the function

$$c^{\mathrm{dom}}_{f,g} : w \longmapsto \begin{cases} g(w) & w \in \mathrm{dom}(f) \\ \uparrow & w \notin \mathrm{dom}(f) \end{cases}$$

is computable.

RANGE CHECK

Let f, g be computable. Then the function

$$c^{\mathrm{ran}}_{f,g} : w \longmapsto \begin{cases} g(w) & \text{if } w \in \mathrm{ran}(f) \\ \uparrow & w \notin \mathrm{ran}(f) \end{cases}$$

is computable.

IMPORTANT REMARK

As opposed to the earlier properties, it's not completely obvious that DOMAIN CHECK & RANGE CHECK by a reasonable model of computation. We need to give an (informal) argument later.

Proof of Theorem

(1) $\Longrightarrow$ (2). Suppose $A = dom(f)$ where f is computable.

Need to show: ψ_A is computable.

$$\underset{\psi_A(w)}{\overset{\parallel}{d_{\sigma\sigma} \circ f(w)}} = \begin{cases} \sigma & \text{if } w \in dom(f) \\ \uparrow & \text{if } w \notin dom(f) \end{cases}$$

So ψ_A is the composition of two computable functions, so computable.

(2) $\Longrightarrow$ (3) Suppose ψ_A is computable.

Need to show ψ_A^* is computable.

Claim: $C^{dom}_{\psi_A, id} = \psi_A^*$.

$$C^{dom}_{\psi_A, id} : v \longmapsto \begin{cases} id(v) = v & \text{if } v \in \overset{\overset{A}{\parallel}}{dom(\psi_A)} \\ \uparrow & \text{if } v \notin A \end{cases}$$

So by DOMAIN CHECK and IDENTITY, we get ψ_A^* is computable.

(3) $\Rightarrow$ (4). Assume ψ_A^* is computable

$$\psi_A^* : v \longmapsto \begin{cases} v & \text{if } v \in A \\ \uparrow & \text{o/w} \end{cases}.$$

Clearly, $\mathrm{ran}(\psi_A^*) = A$.

(4) $\Rightarrow$ (1). Assume $A = \mathrm{ran}(f)$ for f computable.

$$c_{f,id}^{\mathrm{ran}} : v \longmapsto \begin{cases} id(v) & \text{if } v \in \underset{\overset{\parallel}{A}}{\mathrm{ran}(f)} \\ \uparrow & \text{if } v \notin A \end{cases}.$$

By RANGE CHECK, $c_{f,id}^{\mathrm{ran}}$ is computable.

But $\mathrm{dom}(c_{f,id}^{\mathrm{ran}}) = \mathrm{ran}(f) = A$.

q.e.d.

<u>Corollary</u> If M has all of these properties, then K satisfies (1) to (4).

Corollary If M has all of these properties and B is c.e. and $A \leq_m B$, then A is c.e.

Proof. By Theorem ψ_B is computable.

$$\psi_B : v \longmapsto \begin{cases} \sigma & \text{if } v \in B \\ \uparrow & \text{o/w} \end{cases}.$$

Let f be a reduction function from A to B:

f.a. w: $w \in A \longleftrightarrow f(w) \in B$.

Consider $\psi_B \circ f : v \longmapsto \begin{cases} \sigma & \text{if } v \in A \\ \uparrow & \text{if } v \notin A \end{cases}$

So $\psi_A = \psi_B \circ f$ is computable as the composition of two computable functions.

q.e.d.

It remains to give the informal arguments why a notion of computation should allow for properties DOMAIN CHECK and RANGE CHECK.

[We are informally assuming that everything I can do in FORTRAN / PASCAL / C ... can be done by a reasonable model of computation.]

DOMAIN CHECK   Suppose f and g can be done by a computer program.

Find a computer program that does

$$C^{\mathrm{dom}}_{f,g} : \quad v \longmapsto \begin{cases} g(v) & \text{if } v \in \mathrm{dom}(f) \\ \uparrow & \text{o/w} . \end{cases}$$

Algorithm:
1. Put v in storage.
2. Run the f calculation on v. If that never halts, we are undefined.
3. If it halts, then retrieve v from memory.
4. Run the g calculation on v.

$v \in \mathrm{dom}(f) \rightsquigarrow g(v)$ output

$v \notin \mathrm{dom}(f) \rightsquigarrow \uparrow$ output. $C^{\mathrm{dom}}_{f,g}$.

Since we aim for a model of computation s.t. everything programmable in a standard programming language is computable and we checked that $c^{\mathrm{dom}}_{f,g}$ is programmable if f & g are, this constitutes an informal argument for DOMAIN CHECK.

RANGE CHECK If f, g are computable, then so is

$$c^{\mathrm{ran}}_{f,g} : v \longmapsto \begin{cases} g(w) & \text{if } v \in \mathrm{ran}(f) \\ \uparrow & \text{o/w} \end{cases}$$

(1) First write all elements of Σ^* in a canonical order, e.g.,

$\varepsilon, \sigma_0, \sigma_1, \ldots, \sigma_n, \sigma_0\sigma_0, \sigma_0\sigma_1, \sigma_0\sigma_2, \ldots, \sigma_1\sigma_0, \ldots$
$\sigma_0\sigma_0\sigma_0 \ldots$

Let w_i be the i-th word in this order. A computer program can compute $i \longmapsto w_i$.

(2) The bijection $\mathbb{N} \longrightarrow \mathbb{N}^2$ can be explicitly written as

$$\langle i,j \rangle := \frac{(i+j)(i+j+1)}{2} + j .$$

There is a computer program that gives us i and j on input $\langle i,j \rangle$.

$\boxed{\text{Next time}}$ we'll give the algorithm for

ran

cfg.

Tuesday 16 November 2021
11-13.