XVII

AUTOMATA & FORMAL LANGUAGES
Seventeenth Lecture
12 NOVEMBER 2022

M register machine

$$f_{M,k}(\vec{w}) = \begin{cases} v_0 & \text{if computation of } M \text{ with input } \vec{w} \text{ halts with register content } \vec{v} \\ \uparrow & \text{o/w} \end{cases}$$

<u>Def.</u> f is **COMPUTABLE** if there is M, k s.t. $f = f_{M,k}$.

$W_M := \mathrm{dom}(f_{M,1}) \subseteq \mathbb{W}$.

<u>Def.</u> If $X \subseteq \mathbb{W}^k$, then $\chi_X(\vec{w}) := \begin{cases} a & \text{if } \vec{w} \in X \\ \varepsilon & \text{if } \vec{w} \notin X \end{cases}$

THE CHARACTERISTIC FUNCTION OF X

$X \subseteq \mathbb{W}^k$ is **COMPUTABLE** if χ_X is computable.

This allows us to say things like "$L \subseteq \mathbb{W}$ is computable".

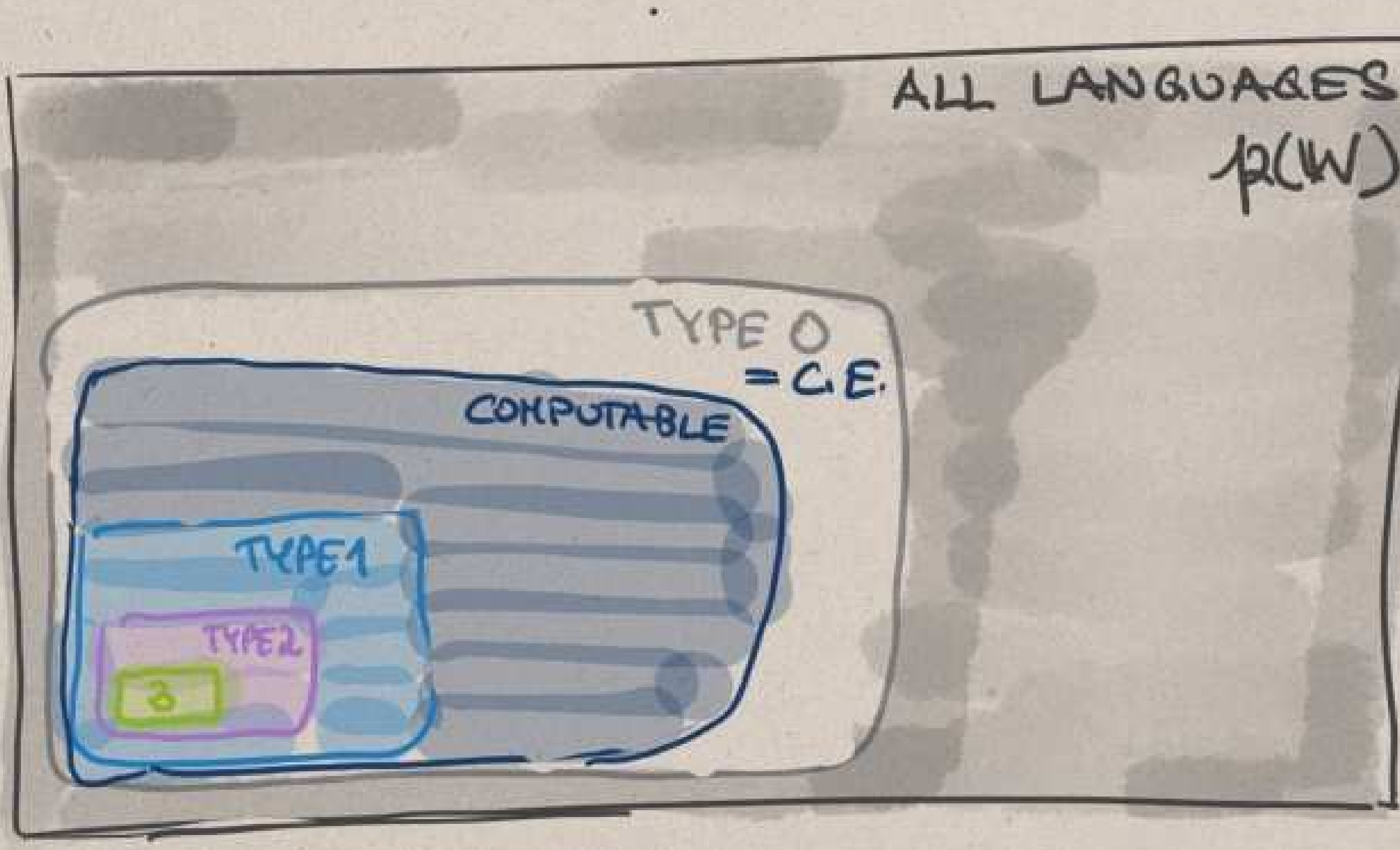
<u>Def.</u> If $X \subseteq \mathbb{W}^k$, then $f : \mathbb{W}^k \dashrightarrow \mathbb{W}$ is called a <u>pseudocharacteristic function</u> of X if $\mathrm{dom}(f) = X$.

ψ_X <u>the</u> <u>pseudocharacteristic function</u> $f(\vec{w}) = \begin{cases} a & \text{if } \vec{w} \in X \\ \uparrow & \text{o/w} \end{cases}$

Def. A set $X \subseteq \mathbb{W}^k$ is called COMPUTABLY ENUMERABLE (C.E.) if ψ_X is computable.

We'll see: every computable set is c.e.
but the converse does not hold.

We'll also see:
$L \subseteq \mathbb{W}$ is c.e. $\iff$ L is type 0.



Prop. 4.11 Let $X \subseteq \mathbb{W}^k$. Then:

(a) X computable $\Longleftrightarrow$ $\mathbb{W}^k \setminus X$ computable.

(b) X is c.e. $\Longleftrightarrow$ $\exists M$ .. $X = \mathrm{dom}(f_{M,k})$

(c) X is computable $\Longrightarrow$ X is c.e.

Proof. We observed that if g, h are computable, then by Case Distinction Lemma, so is

To simplify notation, do $k=1$.

$$f(w) = \begin{cases} g(w) & \text{if } w \neq \varepsilon \\ h(w) & \text{if } w = \varepsilon \end{cases}.$$

	g	h	f
f_1	constant ε	constant a	ε if $w \neq \varepsilon$ a if $w = \varepsilon$
f_2	constant a	constant ε	a if $w \neq \varepsilon$ ε if $w = \varepsilon$
f_3	constant a	$\uparrow$	a if $w \neq \varepsilon$ $\uparrow$ if $w = \varepsilon$

(a) $\chi_{\mathbb{W}^k \setminus X} = f_1 \circ \chi_X$

(b) Let c_a be the constant function with value a and $X = \mathrm{ran}(f)$. Then $\psi_X = c_a \circ f$.

Corrected after the lecture.

(c) $\psi_X = f_3 \circ \chi_X$.

q.e.d.

Theorem Every regular language is computable.

Proof. Let $\mathcal{D} = (\Sigma, Q, \delta, q_0, F)$ be a deterministic automaton s.t. $L = \mathcal{L}(\mathcal{D})$.

$\boxed{\text{Idea}}$ Mimic the behaviour of $\mathcal{D}$ on input $w \in \mathbb{W}$ and output a if $w \in \mathcal{L}(\mathcal{D})$ & ε if $w \notin \mathcal{L}(\mathcal{D})$.

Remark. Automata read forward, RMs read backwards, so we need to reverse the word.

First step Reverse content of reg. 0 into reg. 1.

For each $q \in Q$, the RM will have a set of state Q_q that indicate that we're currently mimicking $\mathcal{D}$ in state q.
While we are in states in Q_q, we will not leave Q_q unless explicitly mentioned.

Second step

Move into a state that lies in Q_{q_0}.

Third Step. Suppose we are in a state $\bar{q} \in Q_q$.

Read the final letter of register 1;
if there is no final letter, i.e.,
register 1 is empty, then

(a) go to Step 4 if $q \in F$
(b) go to Step 5 if $q \notin F$

If it's not empty, say the final letter
is b, remove it and move into
a state lying in $Q_{q'}$
where $q' := \delta(q, b)$.

After that repeat Step 3.

Step 4 Empty reg. 0, add a to reg. 0.
HALT.

Step 5 Empty reg. 0, HALT.

q.e.d.

---

§ 4.4 The shortlex ordering

Goal: Create $<$ on $\mathbb{W}$ s.t.

$$(\mathbb{N}, <) \cong (\mathbb{W}, <).$$

Let us assume that we have total order on Σ: $\Sigma = \{a_0, \dots, a_n\}$

$$a_0 < a_1 < \dots < a_n$$

[Our order will depend on this choice.]

$w, v \in \mathbb{W}$ $w = b_0 \dots b_k$ $v = c_0 \dots c_\ell$

$w < v :\Longleftrightarrow |w| < |v|$ $[k < \ell]$

OR $|w| = |v|$ & the least m s.t. $b_m \neq c_m$ has the property $b_m < c_m$

Called SHORTLEX ORDER

Properties

① Shortlex is a total order on $\mathbb{W}$:

- irreflexive $(w \not< w)$
- transitive $(u<v,\ v<w \longrightarrow u<w)$
- trichotomous $(u<v$ or $v<u$ or $v=u)$

② ε is the least elt of $<$

EXAMPLE $\Sigma = \{0,1\}$ $\quad 0<1$

0	$\boxed{\varepsilon}$
1	0
2	1
3	00
4	01
5	10
6	11
7	000
8	001
9	010
10	011
11	100
12	101
13	110
14	111
15	0000

$2^0 \qquad \boxed{2^0+2^1} \qquad \boxed{2^0+2^1+2^2} \qquad \boxed{2^0+2^1+2^2+2^3}$

Addition on $\mathbb{W}$

$10 + 01 \quad 010$

$5 + 4 = 9$

EXAMPLE

$\Sigma = \{0, 1, 2\}$ $\quad 0 < 1 < 2$

0	ε	1	0	4	00	13	000
		2	1	5	01	14	001
		3	2	6	02	15	002
				7	10	16	010
				8	11	17	011
				9	12	18	012
				10	20		⋮
				11	21		
				12	22		

$1 = 3^0$

$3 = 3^1$

$4 = 3^0 + 3^1$

$9 = 3^2$

$13 = 3^0 + 3^1 + 3^2$

$27 = 3^3$

$40 = 3^0 + 3^1 + 3^2 + 3^3$

ADDING WORDS

$01 + 10 = 22$

$5 + 7 = 12$

MULTIPLYING WORDS

$2 \cdot 02 = 012$

$3 \cdot 6 = 18$

Theorem $(\mathbb{W}, <) \cong (\mathbb{N}, <)$

Shortlex has the same order type as $\mathbb{N}$.

Proof.

"having the same order type" is another expression for "is isomorphic to"

If you fix w, then
$\{v ; v < w\}$
is a finite set of words. Therefore the function

$$\#(w) := |\{v ; v < w\}|$$

is welldefined and it is an isomorphism.

q.e.d.

Theorem (a) $\{(v,w) ; v < w\}$ is computable

(b) $s : \mathbb{W} \longrightarrow \mathbb{W}$ with

$$\#(s(w)) = \#(w) + 1$$

SUCCESSOR FUNCTION

is computable.

Proof. The question Is $|w_i| < |w_j|$ or $|w_i| > |w_j|$ or $|w_i| = |w_j|$?

can be answered by RM.

[Copy i & j into empty registers, remove letters one by one. If one empty before the other, that one is shorter. If not, equal length.

If $|w_i| = |w_j|$, we copy them into empty registers and check & remove one by one until the content is different; then check which one is smaller.
If never different: NO.

q.e.d. (a).

(b) Find the first letter from the back that is not a_n [the largest letter in ordering of Σ]
$\boxed{\text{say } a_i < a_n}$
then switch this to a_{i+1} and fill the rest with as many a_0s as you removed a_ns.
If it turns out that $w = \underbrace{a_n \cdots a_n}_{k \text{ times}}$,

then output $\underbrace{a_0 \cdots a_0}_{k+1 \text{ times}}$.

q.e.d.

HOMEWORK
Note that the construction in (b) is less detailed than previous constructions.
If you feel uncomfortable about this, spell it out in detail.
E.g., "Find the first letter from the back that is not a_n":
→ Read the final letter from reg. 0; if it's a_n, move it to reg. k and continue reading; otherwise ..."