XVI

AUTOMATA AND FORMAL LANGUAGES

Sixteenth Lecture
10 November 2022

From Lecture XIV

Register machines $M = (\Sigma, Q, P)$

Configurations $Q \times \mathbb{N}^{n+1}$

M transforms C into C'

Computation sequence

$C(0, M, \vec{w}) := (q_S, \vec{w})$

$C(i+1, M, \vec{w}) := C'$ if M transforms $C(i, M, \vec{w})$ into C'.

Operations $f: \mathbb{N}^{n+1} \dashrightarrow \mathbb{N}^{n+1}$

Questions $Q = \bigcup_{i \leq k} A_i$; A_i disjoint.

M performs f / M answers Q.

SUBROUTINE LEMMA / CASE DISTINCTION LEMMA

From Lecture XV

Examples of operations performed & questions answered

BASIC
1. Never halt
2. Halt without changing anything
3. Is register i empty?
4. Does register i end with letter a?

MORE
1. Delete the final letter in register i, if it exists.
2. Add a to register i.
3. Empty register i.

4. "Add $w \in W$ to register i"

$w = a_0 \dots a_\ell$

Perform "Add a_0 to register i"

⋮

"Add a_ℓ to register i"

concatenated by Subroutine Lemma.

5. "Replace the reg. content of reg i by the word w"

First empty reg. i;
then add w to register i.

(6) "What is the final letter of reg. i?"

If $|\Sigma| = k+1$, then this question has $k+2$ answers:

$$A_j := \{\vec{w} \,;\, w_i = va_j \text{ some } v\}$$

$$A_\varepsilon := \{\vec{w} \,;\, w_i = \varepsilon\}$$

$$\Sigma = \{a_0, \dots, a_k\}$$

Ask question

"Does reg. i end in a_0?"

If yes, go to state $\overline{q_0}$.

If no, ask question

"Does reg. i end in a_1?"

If yes, go to state $\overline{q_1}$.

⋮

"Does reg. i end in a_k?"

If yes, go to state $\overline{q_k}$.

If no, go to state $\overline{q_\varepsilon}$.

(7) "Copy the final letter of reg. i to reg. j."
(if it exists)

First answer the question
"What's the final letter of i?"
⟿ $\overline{q_\ell}$. (6)

Then add q_ℓ to reg. j. (2)

(8) "Move the final letter from i to j"

First copy by (7),
then remove the final letter by (4).

(9) "Move content of reg. i to reg. j in reverse order"

Repeatedly apply (8) until the question answers state $\overline{q_\varepsilon}$, then halt.

(10) "Move content from i to j"

Let k be an unused empty register.
Move i to k in reverse order. (9)
Move k to j in reverse order. (9)

(11) "Copy content from reg. i to reg j." in reverse order

Do the same process a moving from i to j. in reverse order — (9)

but also in each step write a_ℓ into an unused empty register k.

After the move is done, reverse the content of k back into i.

(12) "Copy content from i to j"

Copy from i to k in reverse order (11)

[where k is empty unused];

move from k to j in reverse order. (9)

(13) "Is the content of reg. i the word w?"

$w = a_0 \dots a_k$

<u>Subroutine</u> S_ℓ:

Answer "Is a_ℓ the final letter of i?"

[empty unused]

If no: q_N

If yes: move final letter to k and run subroutine $S_{\ell-1}$ [if $\ell>0$]

if $\ell=0$ go to q_Y

q_N: Move content of k to i and q_{No}

q_Y: Move content of k to i and q_{Yes}.

§4.3 Computable functions and sets

<u>Remark</u> A lot of computations require the use of <u>scratch space</u>, so we want to reduce the mathematical information that the machine provides to the stuff that's not scratch.

M Register machine $k \in \mathbb{N}$, define

$$f_{M,k} : \mathbb{W}^k \dashrightarrow \mathbb{W}$$

by $f_{M,k}(\vec{w})\uparrow$ iff M does not halt on input $\vec{w}$

$f_{M,k}(\vec{w}) = v_0$ iff M halts on input $\vec{w}$ with halting register content $\vec{v}$.

If M, M' are strongly equivalent,

$f_{M,k} = f_{M',k}$ for all k.

The converse is not true (ES#3).

For the special case $k=1$, we also write
$$W_M := \mathrm{dom}(f_{M,1}).$$

<u>Def.</u> A partial function
$$f: \mathbb{W}^k \dashrightarrow \mathbb{W}$$
is called <u>computable</u> if there is an M s.t. $f = f_{M,k}$.

<u>Observation</u>

1. There are only countably many computable functions.
2. For each computable f, there ∞ many M s.t.
$$f = f_{M,k}.$$
3. Concatenation & Case Distinction Lemma give us that the computable functions are closed under concatenation & case distinction.

So: if $f: \mathbb{W}^k \dashrightarrow \mathbb{W}$ is computable & $g: \mathbb{W} \dashrightarrow \mathbb{W}$ is computable, then $g \circ f$ is computable.

If $Q = \{A_0, A_1\}$ is answered by RM & f, g computable, then $h(\vec{w}) := \begin{cases} f(\vec{w}) & \vec{w} \in A_0 \\ g(\vec{w}) & \vec{w} \in A_1 \end{cases}$ is computable.

Examples

(1) $\mathrm{id} : \mathbb{W} \longrightarrow \mathbb{W}$ IDENTITY

["Do nothing and halt" gives us id.]

(2) $c : \mathbb{W}^k \longrightarrow \mathbb{W}$ CONSTANT

$\vec{w} \longmapsto v$

["Replace content of reg. 0 with v".]

(3) $\pi_i : \mathbb{W}^k \longrightarrow \mathbb{W}$ PROJECTION

$\vec{w} \longmapsto w_i$

[if $i=0$, do nothing;
if $i>0$, empty reg. 0 and copy reg. i to reg. 0.]

Def. Let $X \subseteq \mathbb{W}^k$. We say that f is a characteristic function of X if f is total and $f(\vec{w}) \neq \varepsilon \iff \vec{w} \in X$.

Fix $a \in \Sigma$, we say that f is the characteristic function if

$$f(\vec{w}) := \begin{cases} a & \text{if } \vec{w} \in X \\ \varepsilon & \text{o/w} \end{cases}.$$

Def. $X \subseteq \mathbb{N}^k$ is COMPUTABLE if the characteristic function of X, usually denoted by

$$\chi_X,$$

is computable.