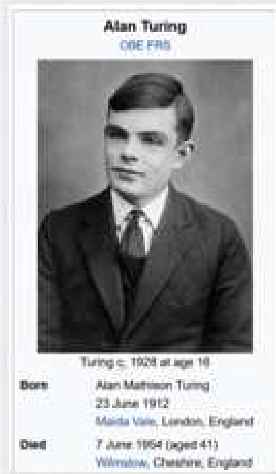
Automata & Formal Languages

XIII : Thirteenth Lecture 12 November 2024

Chapter 4

Aim: The general model of computation



TURING MACHINES
(1936)

REGISTER MACHINES (von Neumann architecture)

- Finitely many storage cells that can be independently accessed and modified.

Five types of operation

$+_0$
$+_1$
$?_0$
$?_1$
$?_\varepsilon$
$-$

Instruction	Interpretation
$+_a(k,q)$	"Add the letter a to the content of register k and go to state q."
$?_a(k,q,q')$	"Check whether the last letter in register k is a; if so, go to state q; otherwise, go to state q'."
$?_\varepsilon(k,q,q')$	"Check whether register k is empty; if so, go to state q; otherwise, go to state q'."
$-(k,q,q')$	"Check whether register k is empty; if so, go to state q; otherwise, remove the final letter of its content and go to state q'".

Table 1: Interpretations of register machine instructions.

REGISTERS are stacks (LIFO storage)

Last in first out

DEFINITIONS

Let Q be a finite set of states.

The following are called Q-instructions:

if $k \in \mathbb{N}$, $q, q' \in Q$

$+_{0}(k, q)$ $\quad ?_{0}(k, q, q')$

$+_{1}(k, q)$ $\quad ?_{1}(k, q, q')$

$?_{\varepsilon}(k, q, q')$

$-(k, q, q')$

In general, accessing any information not at the top of a stack DESTROYS information, but since we have multiple registers, we can copy info somewhere else.

A REGISTER MACHINE $M = (Q, P)$ is a finite set of states with $q_S \neq q_H \in Q$

q_S: START STATE, q_H: HALT STATE

s.t. $P : Q \longrightarrow \mathrm{Inst}_Q$ where Inst_Q is the set of Q-instructions.

__Interpretation__ If the machine is in state q, then it performs the operation $P(q)$.

Note that, since Q is finite, there is a number

$$\underbrace{URI}_{\text{upper register index}}(M) = \max\{k,\ k \text{ occurs in some } P(q)\}$$

Let $n := URI(M)$, then sequences from $Q \times B^{n+1}$ are called

CONFIGURATIONS or SNAPSHOTS

$$(\underset{\substack{\uparrow \\ \text{the STATE} \\ \text{of the} \\ \text{configuration}}}{q}, \underbrace{w_0, \dots, w_n}_{\text{REGISTER CONTENT}})$$

We say that a sequence $C := (q, w_0, ..., w_n) \in Q \times \mathbb{B}^{n+1}$ is a *configuration* or *snapshot* of length $n+1$. In such a configuration, the first entry q is called the *state* of the configuration and the rest is called the *register content* of the configuration. If M is a register machine with upper register index n and C is any configuration of length $m \geq n+1$, then we can define the action of M on C: we say that $\boxed{M \textit{ transforms } C \textit{ to } C'}$ if the following is true:

Case 1. If $P(q) = +_a(k, q')$ and $C' = (q', w_0, ..., w_{k-1}, w_k a, w_{k+1}, ..., w_m)$.

Case 2. If $P(q) = ?_a(k, q', q'')$,

Subcase 2a. $w_k = wa$ for some w and $C' = (q', w_0, ..., w_m)$ or

Subcase 2b. $w_k \neq wa$ for any w and $C' = (q'', w_0, ..., w_m)$.

Case 3. If $P(q) = ?_\varepsilon(k, q', q'')$,

Subcase 3a. $w_k = \varepsilon$ and $C' = (q', w_0, ..., w_m)$ or

Subcase 3b. $w_k \neq \varepsilon$ and $C' = (q'', w_0, ..., w_m)$.

Case 4. If $P(q) = -(k, q', q'')$,

Subcase 4a. $w_k = \varepsilon$ and $C' = (q', w_0, ..., w_m)$ or

Subcase 4b. $w_k = wa$ for some a and $C' = (q'', w_0, ..., w_{k-1}, w, w_{k+1}, ..., w_m)$.

COMPUTATION SEQUENCE of M with input $\vec{w}$

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

$C(k+1, M, \vec{w}) := C'$ if M transforms $C(k, M, \vec{w})$ to C'

M halts on input $\vec{w}$ if there is a k s.t. $C(k, M, \vec{w})$ has the HALT STATE.

The smallest such k is known as the halting time and the register content at this time is the output

[Also say: "M converges" & if it doesn't halt "M diverges"]

Def. M, M' are STRONGLY EQUIVALENT if $\forall k \forall \vec{w}$

1. register content at time k is the same
2. state of M at time k is halting iff state of M' at k is halting

Prop 4.3 Up to strong eq., there are only countably many RM.

Pf. If $|R|$ is n and $|Q| = m$, how many RM can you have with set of states Q?

There are $2\cdot(n+1)\cdot m$ $+$-instructions

$3\cdot(n+1)m^2$ $?$-instructions

$(n+1)m^2$ $-$-instructions

Together $(n+1)m(4m+2)$, so $\left[(n+1)m(4m+2)\right]^m$ RMs.

By Remark, every RM is str. eq. to one of these for some n, m,
so set of RM up to str. eq. is ctble union of finite sets.

q.e.d.

Prop 4.4 (PADDING LEMMA)
For every RM M there infinitely many that are str. eq.

Pf. Show that for each M with m states, there is one str. eq. with $m+1$ states.

Given M, take $\hat{q} \notin Q$ and add $\hat{P}(\hat{q}) := ?_\varepsilon(0, q_\#, q_\#)$.

Since $\hat{q}$ is never reached by the computation seq. $\hat{M} = (Q \cup \{\hat{q}\}, \hat{P})$ is str. eq.

q.e.d.

§4.2 Performing operations & answering questions

<u>Notation for partial functions</u>

$$F : \mathbb{B}^{n+1} \dashrightarrow \mathbb{B}^{n+1}$$

for a partial fn, i.e. there is $\underset{\mathrm{dom}(F)}{A} \subseteq \mathbb{B}^{n+1}$ s.t. $F : A \to \mathbb{B}^{n+1}$

We write $F(\vec{w})\uparrow$ if $\vec{w} \notin \mathrm{dom}(F)$ "diverges"

$F(\vec{w})\downarrow$ if $\vec{w} \in \mathrm{dom}(F)$ "converges"

Start with configuration $(q, \underbrace{w_0, \dots, w_n}_{\vec{w}})$. C

$$P(q) = +_0(k, q') \rightsquigarrow (q', w_0, \dots, w_{k-1}, w_k 0, w_{k+1}, \dots, w_n)$$

[same for 1]

$$P(q) = ?_0(k, q', q'') \rightsquigarrow \begin{cases} (q', \vec{w}) & \text{if } w_k = v0 \\ (q'', \vec{w}) & \text{o/w} \end{cases}$$

[same for 1, ε]

$$P(q) = -(k, q', q'') \rightsquigarrow \begin{cases} (q', \vec{w}) & \text{if } w_k = \varepsilon \\ (q'', w_0, \dots, w_{k-1}, v, w_{k+1}, \dots, w_n) & \text{if } w_k = va \end{cases}$$

new configuration C'

$$\boxed{M \text{ transforms } C \text{ to } C'}$$

Def If $\vec{w}$ is given, we define the COMPUTATION SEQUENCE of M with input $\vec{w}$ as

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

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

We say M performs $F : \mathbb{B}^{n+1} \dashrightarrow \mathbb{B}^{n+1}$ if

$$F(\vec{w})\uparrow \iff M \text{ does not halt on input } \vec{w}$$

$$F(\vec{w})\downarrow \;\&\; F(\vec{w}) = \vec{v} \iff M \text{ halts on input } \vec{w} \text{ and output is } \vec{v}$$

Two examples

NEVER HALT $\quad F$ with $\mathrm{dom}(F) = \emptyset$

ALWAYS HALT WITHOUT CHANGE $\quad F(\vec{w}) = \vec{w}$

$$P(q_S) := ?_{\varepsilon}(0, q_H, q_H)$$