CPSC 421/501 Nov 4

- Recognizing (with a Turing machine):

  - $\text{Acceptance}_{\text{Duck}}$
  - $\text{Acceptance}_{\text{DFA}}$
  - $\text{Acceptance}_{\text{T.m.}}$ universal Turing machine
  - $\text{Acceptance}_{\text{Python}}$

Really § 4.1 - 4.2 [Sip]

For now: (Ch 4)

L is recognizable / decidable

by a multi-tape T.m.,

then also --- with a

one-tape Turing machine

In Ch 7, we'll speak of

polynomial $\underline{\text{time}}$ P. vs NP.

time = ???

length of input

time = "# of operations"

Sorting → steps of comparisons

Python, C, C++, Javascript, ...

→ Assembly Code

well defined operations ...

... ?

Turing machines:

very precise

time = # of steps

# of configurations

space = # of tape cells written to

$$\text{Acceptance}_{\text{Duck}} = \left\{ p\,\sigma_0\,i \;\middle|\; \begin{array}{l} p \text{ is a valid Duck} \\ \text{program that accepts} \\ i \text{ as input} \end{array} \right\}$$

$$\subset \Sigma_{\text{ASCII}}^{*},$$

Σ_{ASCII} has 0–9, a–z, A–Z,

., ;, |, ___

$$\underbrace{\text{separator symbols,}}_{\text{choose } \sigma_0}$$

$\sigma_0 \in \Sigma^*_{ASCII}$ $\qquad \sigma_0 = \langle FS \rangle$

$= \langle \text{file separator} \rangle$

$p\,\sigma_0\,i$ looks like

$$\underbrace{\text{quack|quack3}}_{\text{recognizes}}\ \sigma_0\ \underbrace{abQ}_{\text{is accepted by}}$$

$$\Sigma_{ASCII} \cup \Sigma^3_{ASCII}$$

Build a T.m. that decides $\text{ACCEPTANCE}_{\text{Duck}}$

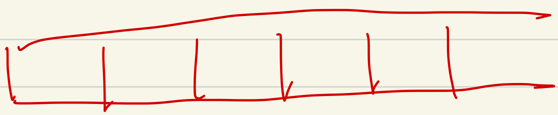
Idea!

count 1

count 2

quack | quack 3 σ_0 a b Q ␣ | ␣ | ␣ ...

Alg: After first σ_0, count how many letters are left before first ␣

scratch paper 

1, 2, 3

"count" each time ∇ → left

count: increment

Step 2 of Alg:

go back to the start, see

if string before first

"σ_c" is a valid Duck

program and has the number

$k = |i| =$ length of i

$=$ length of the input

Acceptance_{DFA}

$\mathbb{N} = \{1, 2, \ldots\}$ infinite countable

Power($\mathbb{N}$) uncountable

Power(Power($\mathbb{N}$))

Power( )

larger and larger

Q finite set

So even if $Q = \{q\}$,

q could be in

$Power(Power(Power(\mathbb{N})))$

vastly infinite

Introduce:

DFA, $M = (Q, \Sigma, \delta, q_0, F)$

input $i \in \Sigma^*$

$\langle M, i \rangle$ description of M and i

① Assume $\Sigma = \{1, 2, \ldots, n_\Sigma\}$

$n_\Sigma \in \mathbb{N}$

So $\cancel{\Sigma = \{a, b\}}$

$\Sigma = \{1, 2\}$

If Σ is not of this form,

$$\Sigma = \{\sigma_1, \sigma_2, \sigma_3\}$$

$$\begin{array}{ccc} \sigma_1 & \sigma_2 & \sigma_3 \\ \updownarrow & \updownarrow & \updownarrow \\ 1 & 2 & 3 \end{array}$$

Give a bijection between Σ and $\{1, 2, \ldots, n_\Sigma\}$

Next assume

$$Q = \{1, 2, \ldots, n_Q\}$$

$$n_Q \in \mathbb{N}$$

Define: A DFA $= (Q, \Sigma, \delta, q_0, F)$ is standardized if

for some n_Q, n_Σ,

$$Q = \{1, 2, \ldots, n_Q\}$$

$$\Sigma = \{1, 2, \ldots, n_\Sigma\}$$

Then, given

$$\text{DFA} = (\{1, \dots, n_Q\}, \{1, \dots, n_\Sigma\}, \delta, q, F)$$

$$= M$$

$$\langle M \rangle$$ over some fixed alphabet

$$\Sigma_{\text{DFA description}} = \{0, 1, \dots, 9, \quad , \quad , \quad \}$$

Say

$$m = (\underset{Q}{\{1,2,3\}}, \underset{\Sigma}{\{1,2\}}, \delta, q_0, F)$$

$$\delta : Q \times \Sigma \to Q$$

here

$$\delta : \{1,2,3\} \times \{1,2\} \to \{1,2,3\}$$

$$q_0 \in \{1,2,3\} = Q$$

$$F \subset \{1,2,3\} = Q$$

$|Q|$ $|\Sigma|$

$$\langle m \rangle = \boxed{3 \# 2 \# \delta \text{ values} \ldots,}$$

One convention

$\delta(1,1) \in \{1,2,3\}$

$\delta(1,2) \in \{1,2,3\}$

$\delta(2,1) \in$ — "

$\delta(2,2) \in$ — "

$\delta(3,1) \in$ "

$\delta(3,2) \in$ "

So ~~to~~ to describe δ:

$\delta(1,1) \# \delta(1,2) \# \ldots \#$

q_0, F