CPSC 421/501 November 6, 2024

- Standardized DFA's, Turing m.'s

- Acceptance$_{\text{DFA}}$

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

- Acceptance$_{\text{Python}}$

We are all T.m. project managers....

Homework: Code snippet to build a counter.

Idea:

$$\Sigma^* \xrightarrow{\text{TM}} \begin{cases} \text{accepts} \\ \text{reject} \\ \text{loops} \end{cases}$$

Homework: $\Sigma = \{0, 1\}$

Input

0	⊔	⊔	...

OR

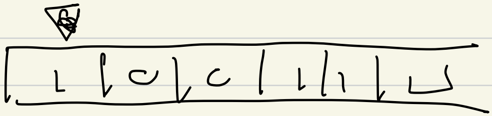
1	⊔	⊔	...

OR

1	0	0	1	1	⊔

Goal: (always accept)

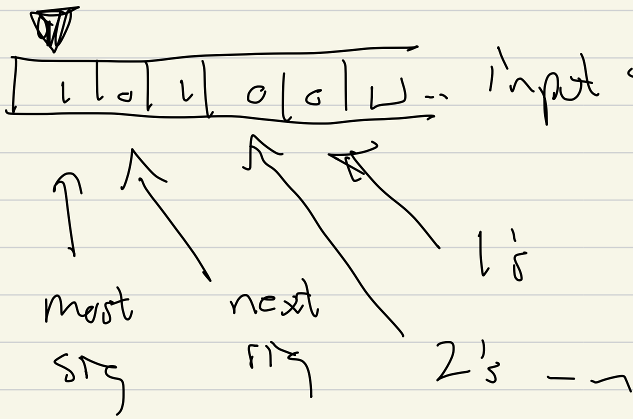
Tape should end with



input

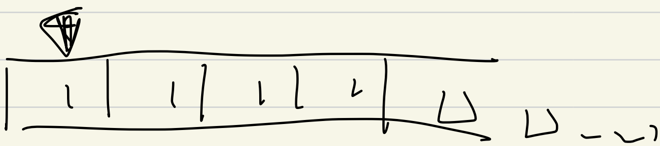
$$
\begin{array}{r}
10011 \\
+\quad 1 \\
\hline
10100
\end{array}
$$

binary

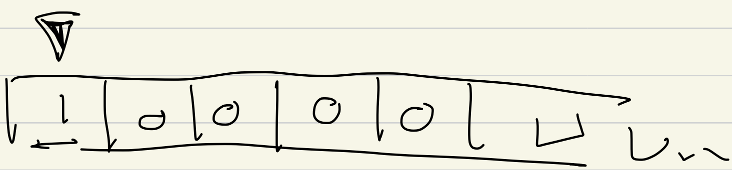


input +1

but input



output



For us, T.m.

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$$

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

not in [Sip], but

$$\mathcal{L} : Q \times \Gamma \to \left\{\begin{array}{l}\text{you're at the left edge} \\ \text{you're not}\end{array}\right\}$$

TM now:

$$(Q, \Sigma, \_\_\_\_\_, \mathcal{L})$$

$$\delta : Q \times \Gamma \times \{\quad\}$$

A DFA $(Q, \Sigma, \delta, q_0, F)$

is standardized if

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

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

for some $n_Q, n_\Sigma \in \mathbb{N} = \{1, 2, \ldots\}$

(optionally, may as well insist that

$q_0 = 1 \in Q$)

$F \subset Q = \{1, \ldots, n_Q\}$

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

So

$$\delta : \{1, \ldots, n_Q\} \times \{1, \ldots, n_\Sigma\}$$

$$\longrightarrow \{1, \ldots, n_Q\}$$

Usually: could write

$\delta(1,1) = 27$

$n_Q \geq 311$

$\delta(1,2) = 311$

$\vdots$

$n_Q = 411$

$n_\Sigma = 5$

n_Q

$$411 \# 5 \# \underbrace{27}_{\delta(1,1)} \# \underbrace{311}_{\delta(1,2)} \# \ldots$$

In this way,

each standardized DFA is
a string of $\{0, 1, \ldots, 9, \#\}$

Alternatively:

$5\underbrace{\#27}\#311$

$\#\#1\#1\#27\#\#1\#1\#311\#\# \_\_ \_\_$

TM!

$(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

input alphabet

tape alphabet

standardized if

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

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

$\Gamma = \{1, \ldots, n_\Gamma\}$

$\Sigma \subset \Gamma$

$\{1, \ldots, n_\Sigma\} \subset \{1, \ldots, n_\Gamma\}$

$\sqcup \longrightarrow n_\Sigma + 1$

$q_0 = 1 \in Q = \{1, \dots, n_Q\}$

$q_{acc} = 2$, $q_{rej} = 3$

(?) could $q_0 = q_{acc}$ yes, but it's not important.

$$\text{Acceptance}_{DEA} = \left\{ \langle M, i \rangle \;\middle|\; \begin{array}{l} M \text{ is a standard DEA,} \\ i \text{ is input} \end{array} \right\}$$

$\Sigma = \{1, \dots, n_\Sigma\}$, $i \in \Sigma^*$

e.g. $i = 231$ $\langle i \rangle = \#2\#3\#1$

$$\text{Acceptance}_{T.m} = \{ \langle M, i \rangle \mid M \text{ is a standardized T.m., } i \text{ is input} \}$$

AND NO CHEATING ...

Example:

$$\langle M, i \rangle$$

$$\delta$$

3 # 7 # 21 # —~ ### 1 # 6 # 4 #

$\uparrow \quad \uparrow \quad \uparrow$

$|Q| \quad |\Sigma| \quad |\Gamma|$

7 # 2 # 3 #

0 or 1

if M halts on input 1

if M does not

$i = 184723$

$\langle M, i \rangle$ is not entirely precise, until you make it precise

[Sip] is a bit "light" on these details ...

Can we recognize / decide

1. Acceptance_{DFA}

2. $\text{Acceptance}_{T.M}$

3. $\text{Acceptance}_{Python}$, $\Sigma_{ASCII} = \{1, 2, \ldots, 128\}$

$0 \quad 1 \quad \ldots \quad 111111$

Acceptance DFA's

$$2 \# 3 \# \overset{\delta}{\text{---}} \# 1 \# 1001 \ldots 1$$

$|Q|$ $|\Sigma|$ q_0 $F \subset Q$

$$\underbrace{\#\# 1 \# 2 \# 2 \# 3}_{i} \# 1$$

Acceptance TM

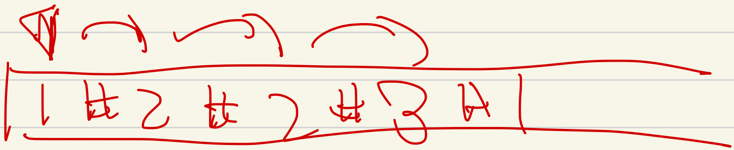
$\langle M, i \rangle$ $\underbrace{\quad}_{}$ $\delta, q_0, q_{acc}\, q_{rej}$

$3 \# 100 \# 105 \# \text{---} \# - \#$ blah blah

$\underbrace{\quad}_{\text{input}} \rightarrow$ blah

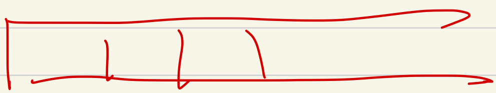
2nd tape:

code-human
code-bot



write input on tape 2

tape 3



current state

Acceptance$_{T.m.}$ on input $\langle M, i \rangle$

"we simulate M on i and return the result if we don't loop.