CS 205
Lecture #4
12/01/21

Recap
- Strings, Languages
- DFA M
- Language accepted by M, $L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$
- Regular language: L is regular if there is a DFA M s.t. $L = L(M)$
- Examples:

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

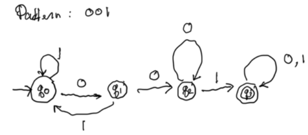
To show that a given language L is regular it is enough to come up with an M s.t. $L = L(M)$.

Def: Say that u is a substring of v, if there are strings w_1, w_2 s.t. $v = w_1 u w_2$.

Ex $\Sigma = \{0,1\}$
L = set of all strings that contain 001 as a substring
- 001, 10011, 0011, ... belong to L
- $11100 \notin L$

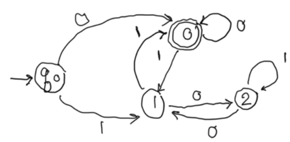
Show that L is regular.

Pattern: 001



Ex $L = \{w \in \{0,1\}^* \mid w$ is the binary representation of a multiple of $3\}$

$L = \{0, 11, 110, 1001, \ldots\}$



Ex $\Sigma = \{0,1\}$
$L_n = \{w \in \Sigma^* \mid n^{th}$ char from end of w is $1\}$

e.g. $L_3 = \{100, 101, 110, 111, 0100, \ldots\}$

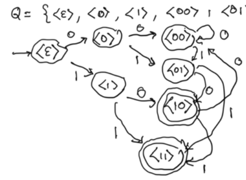
Show that L_n is regular.

Idea $M_n = (Q, \Sigma, \delta, q_0, F)$
A state in Q remembers the last n characters of the input string w

Formally,
$Q = \{\langle w \rangle \mid w \in \Sigma^*$ and $|w| \leq n\}$ — Why?
$|w|$: length of w

$$\delta(\langle w \rangle, a) = \begin{cases} \langle wa \rangle & \text{if } |w| < n \\ \langle w_2 w_3 \ldots w_n a \rangle & \text{if } w = w_1 w_2 \ldots w_n \end{cases}$$

e.g. $n = 2$
$Q = \{\langle\varepsilon\rangle, \langle 0 \rangle, \langle 1 \rangle, \langle 00 \rangle, \langle 01 \rangle, \langle 10 \rangle, \langle 11 \rangle\}$



$q_0 = \langle\varepsilon\rangle$

$F = \{\langle 1 w_2 w_3 \ldots w_n \rangle \mid w_i \in \{0,1\}$ for $2 \leq i \leq n\}$

Prop: Any DFA recognizing L_n must have at least 2^n states.

Proof: By contradiction. Assume some DFA M accepting L has less than 2^n states.

Since the no. of strings of length n is 2^n, there must be two distinct strings w_0 and w_1 of length n s.t. $\delta(q_0, w_0) = \delta(q_0, w_1)$. — Why?