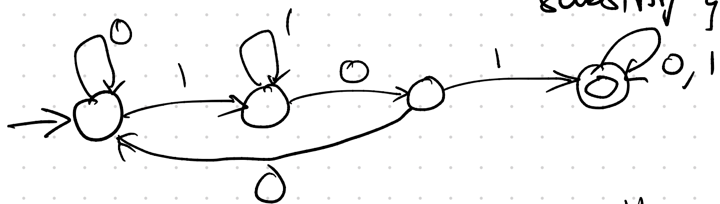
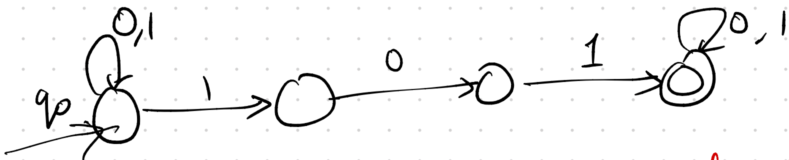
Lecture 2: Nondeterministic Finite Automata (NFA)

Deterministic Finite Automata (DFA) for

$A_{101} = \{ x \in \{0,1\}^* \mid x \text{ contains } 101 \text{ as a substring} \}$



Suppose we could "guess" when the substring begins



Nondeterministic FA

Nondeterministic action on input 1 and start state

When faced with nondeterministic choice, machine forks a branch for each choice "COMPUTATION TREE"

Comment: Nondeterminism not a realistic model

But important conceptually

Eg. [P vs NP question]

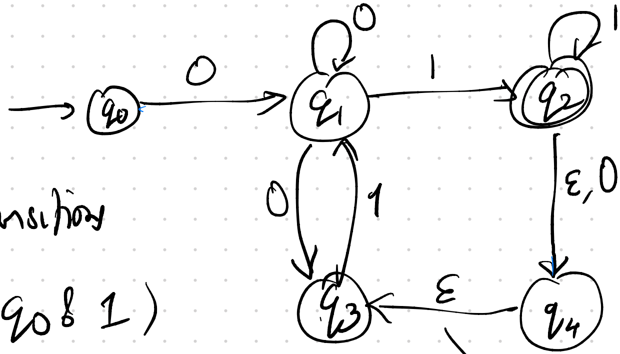
THINK ABOUT DFA/NFA for language

$T = \{ x \in \{0,1\}^* \mid 4^{th} \text{ symbol from the end is a } 1 \}$

$(001001 \in T$

$10110 \notin T)$

Example:



001 - accepted

01011 - accepted

00100 - rejected

→ ε-transitions

$(\delta^*(q_2, 0) = \{q_4, q_3\})$

- Not all transitions present (e.g. q_0 & 1)
- Multiple transitions with some label
  e.g. q_1 and 0 — can go to q_3 or stay at q_1
- Free ε-transitions — move for free without reading next symbol

Accept / reject defn (informal)

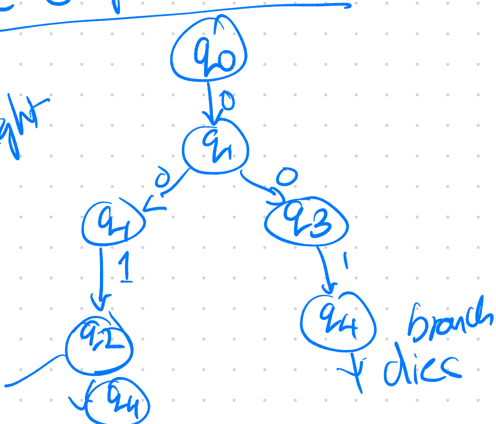
- Input string w is accepted if $\exists$ a valid path labeled w that leads from start state to some final state

  (can take ε's as needed)

- w rejected otherwise.

Visualize Computation tree   00100 (above NFA)

Note some branches might die



Check if some leaf is a final / accept state

NFA formal definition

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

states (Q), input alphabet (Σ), transition fn (δ), start state (q_0), final/accept states (F)

$$(\delta : Q \times \Sigma \longrightarrow Q) \text{ for DFA}$$

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \longrightarrow \mathcal{P}(Q) \rightarrow \text{Power set of } Q.$$

(transitions to 0, 1, or more states on each curr state + input symbol incl. ϵ)

Defn of acceptance:

- For $q \in Q$, define ϵ-closure(q) to be all states that can be reached from q using only a seq. of ϵ-transitions.
  (also $q \in \epsilon\text{-closure}(q)$)
- Define $\delta^*(q, a) = \bigcup_{r \in \delta(q,a)} \epsilon\text{-closure}(r)$

Defn of NFA N accepting $w \in \Sigma^*$, $|w| = n$

w is acc. by N if $\exists$ seq. of states $r_0, r_1, \ldots, r_n$ st

- $r_0 \in \epsilon\text{-closure}(q_0)$ $\quad w = w_1 w_2 \dots w_n$
- $r_i \in \delta^*(r_{i-1}, w_i)$ $\quad$ for $i = 1, 2, \dots n$
- $r_n \in F$

<u>Theorem</u>: L accepted by an NFA $\iff$ L is accepted by some DFA

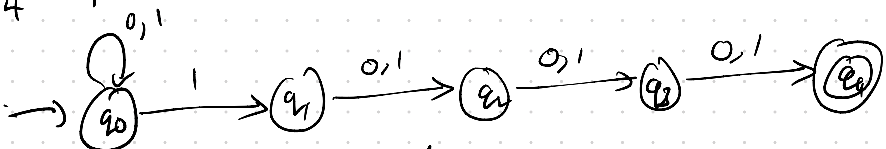
<u>Pf</u>. $\Leftarrow$ obvious

$\Rightarrow$ "subset construction" — Keep track of all states one can be in with a bigger state space

- Exponential blow up in # states
- Necessary

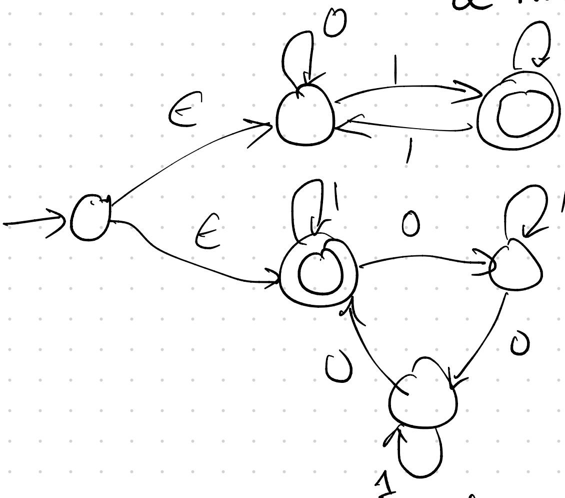
<u>Draw some example NFAs</u>

- $L_4 = \{ x \mid 4^{th}$ bit from end is a $1 \}$



- 5 states
- DFA needs 16 states

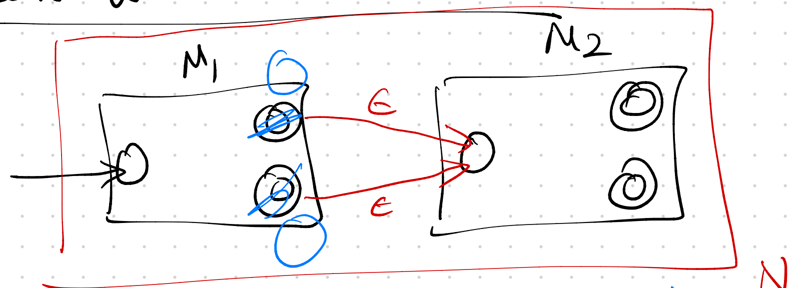
• $B = \{ x \mid x$ has odd # 1's or a multiple of 3 no. of 0's $\}$



"Guess which DFA to run at the beginning"

- Above shows closure under union of regular languages

Closure under concatenation $L_1 L_2$



Exercises: - Show NFAs for L^*

- $L^R = \{ x^R \mid x \in L \}$ reverse of x

NFA to DFA conversion

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

$D = (Q', \Sigma, \delta', q_0', F')$

- $Q' = \mathcal{P}(Q)$ (power set of Q also denoted 2^Q)
- $q_0' = \epsilon\text{-closure}(q_0) \subseteq Q$
- $F' = \{ R \subseteq Q \mid R \cap F \neq \emptyset \}$ (set of states that includes at least one final state from F)
- $\delta'(\underbrace{R}_{\subseteq Q}, \underbrace{a}_{\in \Sigma}) = \bigcup_{r \in R} \delta^*(r, a)$

Exercise: Formally prove that $w \in L(N)$ ⟸ set of strings accepted by N.
$\iff w \in L(D)$

Exercise. Given NFA N & $x \in \Sigma^*$, give an algo to tell if N accepts x in time $O(|N|\,|x|)$ → # states + # transitions in N