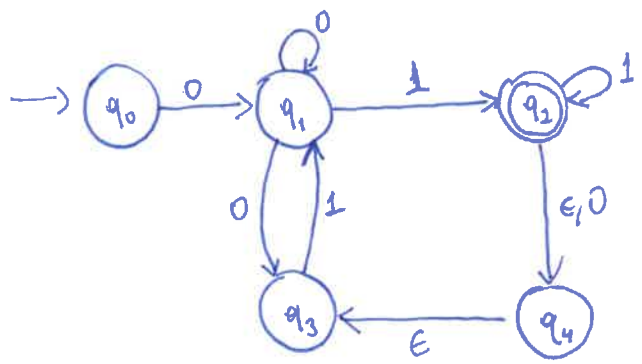
15-252 Spring 2018
Non-Deterministic Finite Automata (NFAs)

Intro:

- non-determinism important concept in CS. Does it give extra power? P vs. NP
  (NON-DET ≠ RANDOMIZATION)
- many closure properties of reg. languages can be proved very easily with NFAs.
  (often math is all about coming up with the right definitions.)

1 

4

001 accepts
01011 accepts
00100 rejects

Comparison with DFAs:

2

- ~~trans~~ not all possible transitions present.
- multiple transitions with same label from a state.
- ϵ transitions.

Rule for accepting/rejecting:

3

<u>Accept</u> w if there is a "valid" path that leads to an accepting state.

<u>Reject</u> w if all paths lead to a rejecting state.

Example computation trace

	0	0	1	1	0
q_0	q_1	q_1	q_2	q_2	q_4
		q_3	q_1	$\cancel{q_2}$	
			q_4		
			q_3	q_1	q_1
				q_4	
				q_3	
				$\cancel{q_4}$	
				$\cancel{q_3}$	
					q_3
					$\cancel{q_3}$

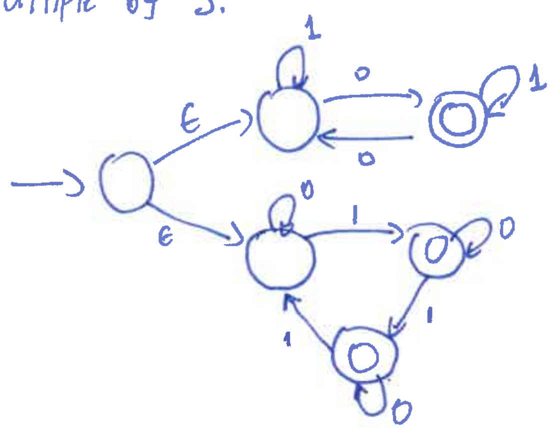
Def: An NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q finite non-empty set (set of states)
- Σ " " " (alphabet)
- $\delta : Q \times \Sigma \cup \{\epsilon\} \to \mathcal{P}(Q)$ (transition function)
- $q_0 \in Q$ (starting state)
- $F \subseteq Q$ (set of accepting states)

Exercise: Define formally the notion of an NFA accepting a string.

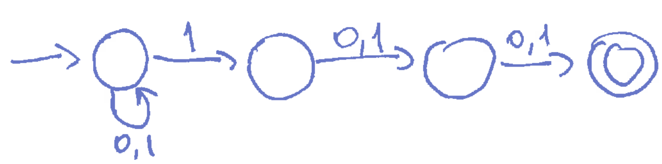
Drawing NFAs

All binary strings that have either odd # 0's or #1's is not a multiple of 3.



Do THIS LAST!

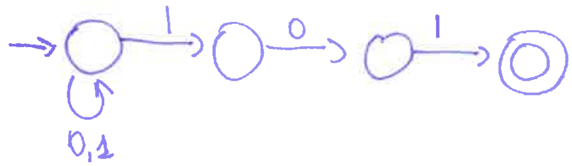
$L = \{ w \in \{0,1\}^* : |w| \geq 3$ and $w_{n-2} = 1$ where $n = |w| \}$



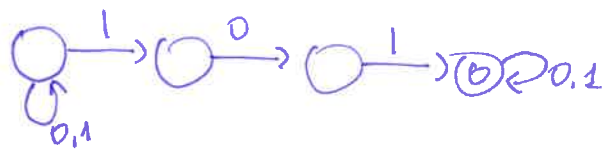
$L_k = \{ w \in \{0,1\}^* : |w| \geq k$ and $w_{n-(k-1)} = 1$ where $n = |w| \}$

Best DFA needs 2^k states.

L = all binary string that end with 101.



L = all binary strings that contain 101 as a substring.



Thm : NFA ≡ DFA (any language recognized by a NFA is regular)

Proof Sketch : Threading idea works.

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. (Suppose N has no ϵ-transitions.) Construct DFA $M = (Q', \Sigma, \delta', q_0', F')$ as follows:

$Q' = \mathcal{P}(Q)$

$\delta' : Q' \times \Sigma \to Q'$, $\forall S \in Q', a \in \Sigma$, $\delta'(S,a) = \bigcup_{s \in S} \delta(s,a)$

$q_0' = \{q_0\}$

$F' = \{S \in Q' : S \cap F \neq \emptyset\}$

---

if N has ϵ-transitions, define, for $S \in Q'$, $E(S) = \{q \in Q : q$ can be reached from a state in S using zero or more ϵ-transitions$\}$

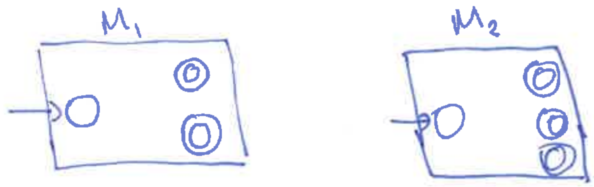
$\delta''(S,a) = E(\delta'(S,a))$

$q_0'' = E(\{q_0\})$

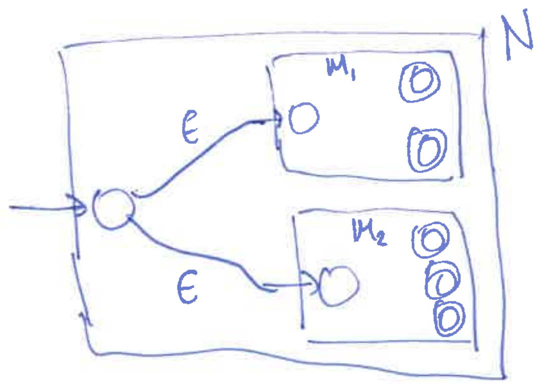
COMMENT: # STATES IN DFAs VS NFAs

Closure Properties of Regular Languages

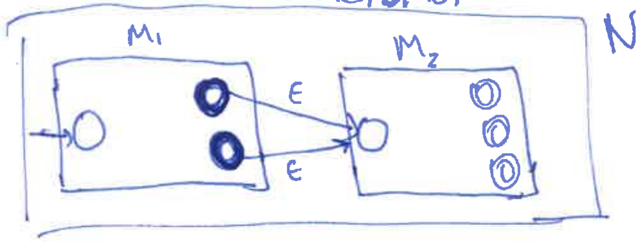
Closure under union:



Build NFA

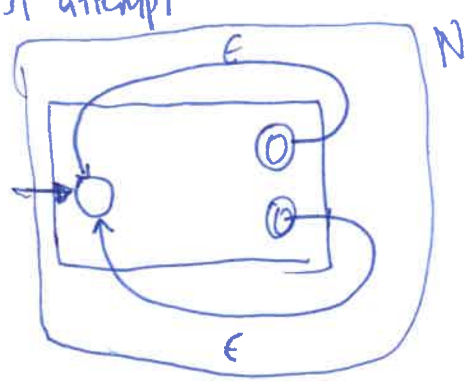


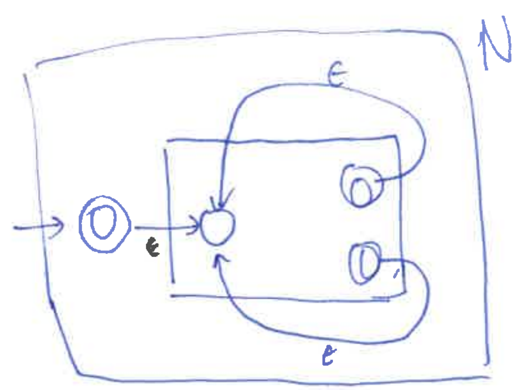
Closure under concatenation



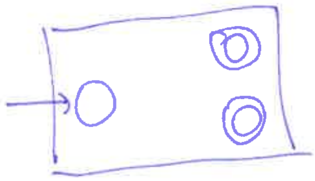
Closure under star

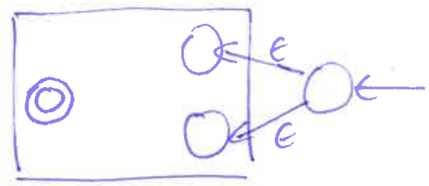
First attempt





Closure under reversal





reverse every transition.