COURSE: THEORY OF AUTOMATA COMPUTATION

TOPICS TO BE COVERED

- Equivalence of DFA and NFA
- NFA with ε moves

EQUIVALENCE OF DFAS AND NFAS

- Do DFAs and NFAs accept the same *class* of languages?
  - Is there a language L that is accepted by a DFA, but not by any NFA?
  - Is there a language L that is accepted by an NFA, but not by any DFA?

- Observation: Every DFA is an NFA.

- Therefore, if L is a regular language then there exists an NFA *M* such that L = L(*M*).

- It follows that NFAs accept all regular languages.

- But do NFAs accept more?

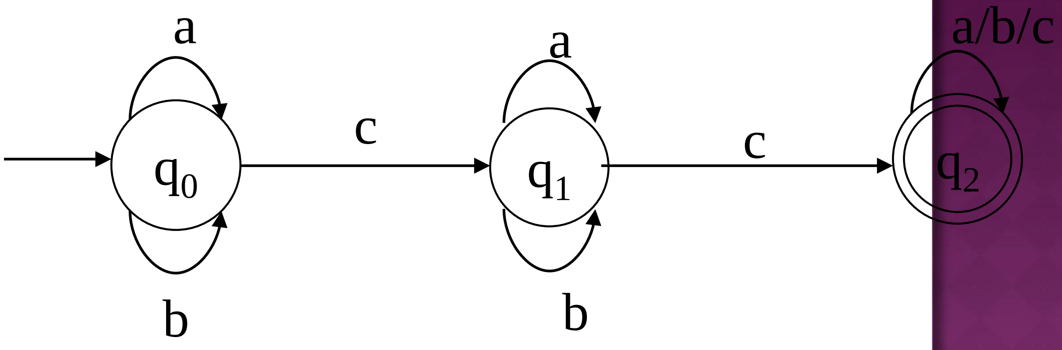
- Consider the following DFA: 2 or more c's

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

Start state is q_0

$F = \{q_2\}$

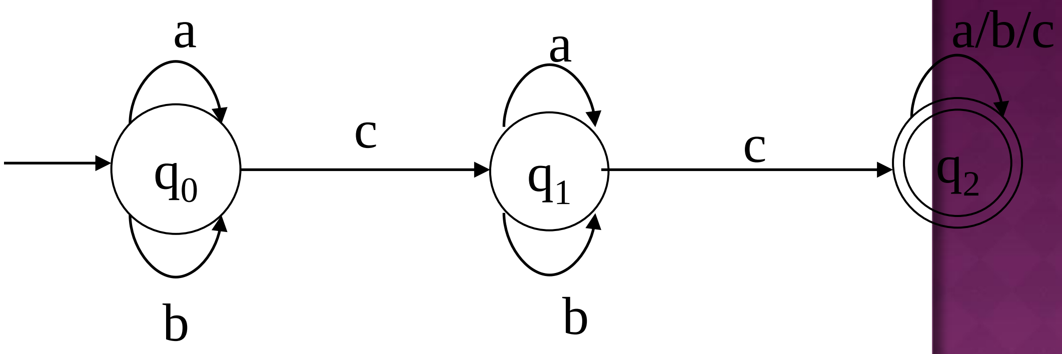


δ:

	a	b	c
q_0	q_0	q_0	q_1
q_1	q_1	q_1	q_2
q_2	q_2	q_2	q_2

- An Equivalent NFA:

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b, c\}$
Start state is q_0
$F = \{q_2\}$



δ:

	a	b	c
q_0	$\{q_0\}$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$

- **Lemma 1:** Let *M* be an DFA. Then there exists a NFA *M'* such that L(*M*) = L(*M'*).

- **Proof:** Every DFA is an NFA. Hence, if we let *M'* = M, then it follows that L(*M'*) = L(*M*).

  The above is just a formal statement of the observation from the previous slide.

- **Lemma 2:** Let M be an NFA. Then there exists a DFA M' such that L(M) = L(M').

- **Proof:** (sketch)

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

Define a DFA $M' = (Q', \Sigma, \delta', q'_0, F')$ as:

$Q' = 2^Q$
$= \{Q_0, Q_1, \ldots,\}$

Each state in M' corresponds to a subset of states from M

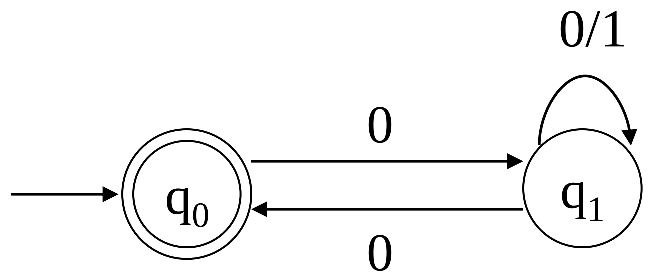
where $Q_u = [q_{i0}, q_{i1}, \ldots q_{ij}]$

$F' = \{Q_u \mid Q_u$ contains at least one state in F$\}$

$q'_0 = [q_0]$

$\delta'(Q_u, a) = Q_v$ iff $\delta(Q_u, a) = Q_v$

- Example: empty string or start and end with 0



$Q = \{q_0, q_1\}$

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

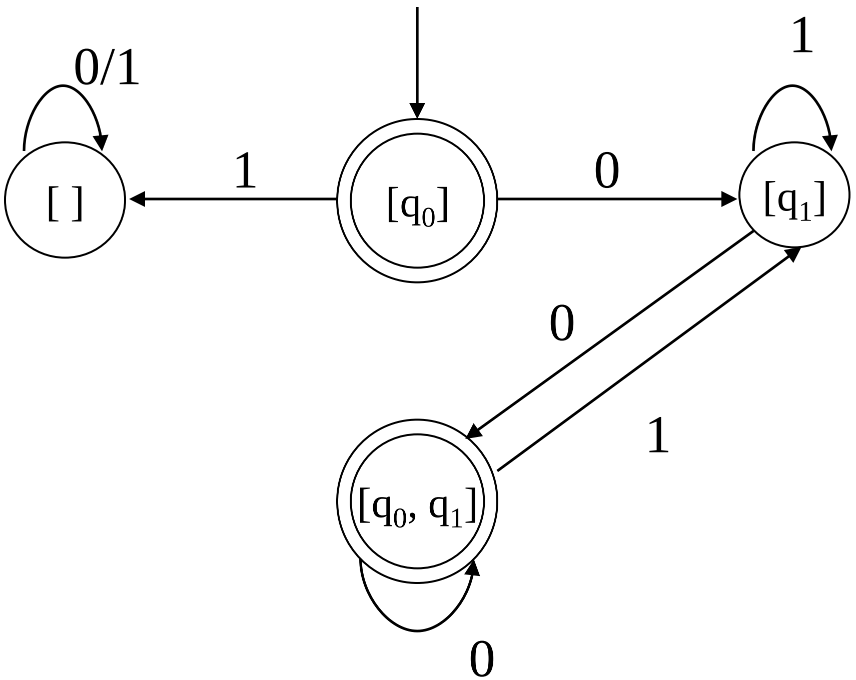
Start state is q_0

$F = \{q_1\}$

δ:

	0	1
q_0	$\{q_1\}$	$\{\}$
q_1	$\{q_0, q_1\}$	$\{q_1\}$

- Construct DFA M' as follows:



$\delta(\{q_0\}, 0) = \{q_1\}$	=>	$\delta'([q_0], 0) = [q_1]$
$\delta(\{q_0\}, 1) = \{\}$	=>	$\delta'([q_0], 1) = [\ ]$
$\delta(\{q_1\}, 0) = \{q_0, q_1\}$	=>	$\delta'([q_1], 0) = [q_0, q_1]$
$\delta(\{q_1\}, 1) = \{q_1\}$	=>	$\delta'([q_1], 1) = [q_1]$
$\delta(\{q_0, q_1\}, 0) = \{q_0, q_1\}$	=>	$\delta'([q_0, q_1], 0) = [q_0, q_1]$
$\delta(\{q_0, q_1\}, 1) = \{q_1\}$	=>	$\delta'([q_0, q_1], 1) = [q_1]$
$\delta(\{\}, 0) = \{\}$	=>	$\delta'([\ ], 0) = [\ ]$
$\delta(\{\}, 1) = \{\}$	=>	$\delta'([\ ], 1) = [\ ]$

- **Theorem:** Let L be a language. Then there exists an DFA M such that L = L(M) iff there exists an NFA M' such that L = L(M').

- **Proof:**

  (if) Suppose there exists an NFA M' such that L = L(M'). Then by Lemma 2 there exists an DFA M such that L = L(M).

  (only if) Suppose there exists an DFA M such that L = L(M). Then by Lemma 1 there exists an NFA M' such that L = L(M').

- **Corollary:** The NFAs define the regular languages.

- Note: Suppose R = {}

$$\begin{aligned}\delta(R, 0) &= \delta(\delta(R, \varepsilon), 0) \\ &= \delta(R, 0) \\ &= \bigcup_{q \in R} \delta(q, 0) \\ &= \{\} \qquad \text{Since } R = \{\}\end{aligned}$$

- Exercise - Convert the following NFA to a DFA:

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
Start state is q_0
$F = \{q_0\}$

δ:	0	1
q_0	$\{q_0, q_1\}$	$\{ \}$
q_1	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$

NFAS WITH E MOVES

- An NFA-ε is a five-tuple:

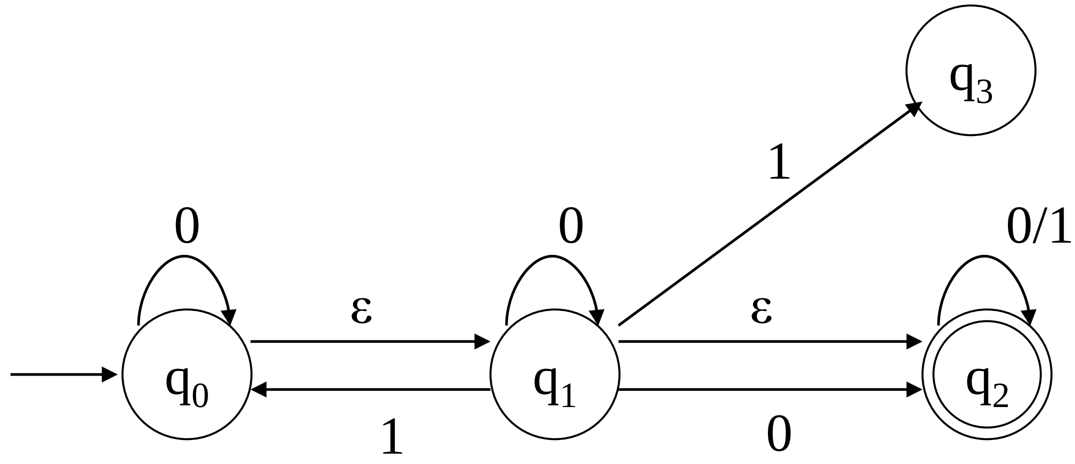
  $M = (Q, \Sigma, \delta, q_0, F)$

  Q A finite set of states
  Σ A finite input alphabet
  q_0 The initial/starting state, q_0 is in Q
  F A set of final/accepting states, which is a subset of Q
  δ A transition function, which is a total function from $Q \times \Sigma \cup \{\varepsilon\}$ to 2^Q

  $\delta: (Q \times (\boldsymbol{\Sigma} \cup \{\boldsymbol{\varepsilon}\})) \rightarrow 2^Q$
  δ(q,s) -The set of all states p such that there is a transition labeled a from q to p, where a is in $\Sigma \cup \{\varepsilon\}$

- Sometimes referred to as an NFA-ε other times, simply as an NFA.

- Example:



δ:	0	1	ε
q_0	$\{q_0\}$	$\{\ \}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_0, q_3\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$	$\{\ \}$
q_3	$\{\ \}$	$\{\ \}$	$\{\ \}$

- A string $w = w_1w_2\ldots w_n$ is processed as $w = \varepsilon^*w_1\varepsilon^*w_2\varepsilon^* \ldots \varepsilon^*w_n\varepsilon^*$

- Example: all computations on 00:

$$\begin{array}{cccc} & 0 & \varepsilon & 0 \\ q_0 & q_0 & q_1 & q_2 \end{array}$$

: