# Session 8

Nondeterministic Finite Automata

# Nondeterministic FA

- Motivation
- Concept of non-determinism
- Definition of NFA
- Extended Transition Function for NFA
- Acceptance by a NFA

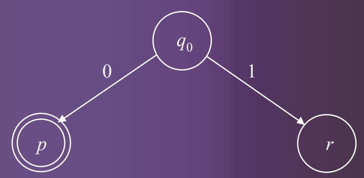
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

## Motivation

- Design a DFA accepting: **$\{11,110\}^*\{0\}$**

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

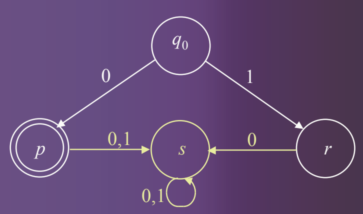
- Let's proceed one symbol at a time: $\{11,110\}^*\{0\}$




- The null string $\Lambda$ is not in $L$, and $q_0$ is not an accepting state
- But 0 is in $L$, and it should take us to an accepting state
- 1 is initial symbol in both 11 and 110, the constituent words of the prefix of strings in the language (before the last 0).

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

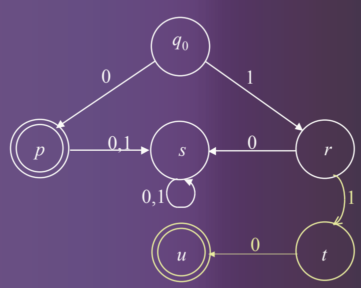
- Design a DFA accepting: $\{11,110\}^*\{0\}$




- $L$ does not contain:
  - Strings starting with 0
  - Strings starting with 10
- We have a sink or dead state!

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

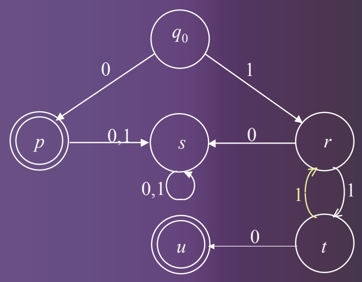
- Design a DFA accepting: $\{11,110\}^*\{0\}$




- In state $r$ we have seen one 1, and we need 10

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

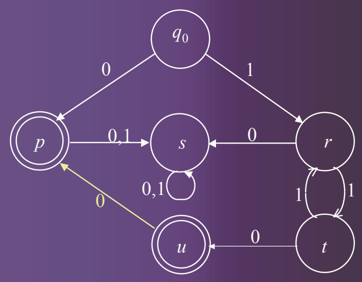
- Design a DFA accepting: {11,110}*{0}




- But we also need to consider repetitions of 11
- The state *r*: having seen an odd sequence of 1's



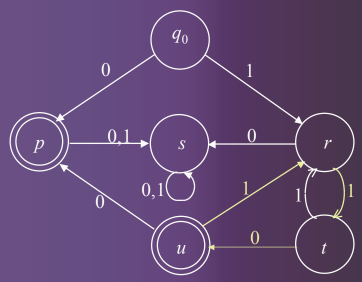
- Design a DFA accepting: {11,110}*{0}




- But also, after 1, we can accept with 100



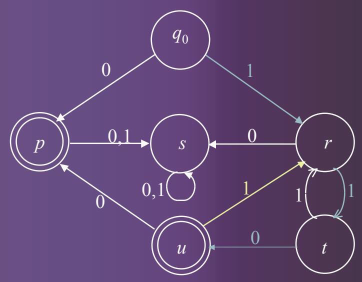
- Design a DFA accepting: {11,110}*{0}




- But, the string 110 before the last 0 can also appear *n* times!



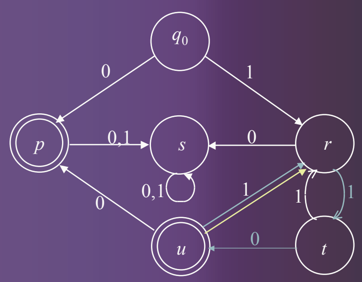
- Design a DFA accepting: {11,110}*{0}




- State *r* also represents having seen the first 1 after 110, the first time



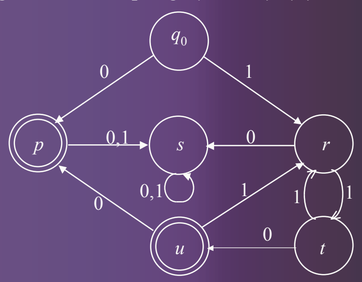
- Design a DFA accepting: {11,110}*{0}




- State *r* also represents having seen the first 1 after 110, or after *n* number of times!

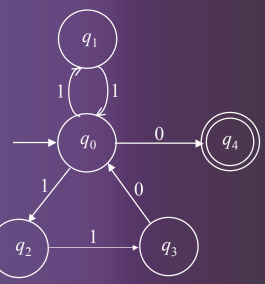


- Design a DFA accepting: {11,110}*{0}




- For more complicated *RE*
  - Finding the equivalent FA can be quite laborious!
  - Making sure it is right: very complicated!

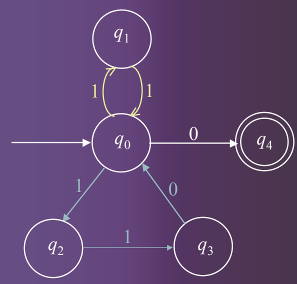
- But what about this, for accepting: **{11, 110}*{0}**




- We can read the expression in the diagram directly!



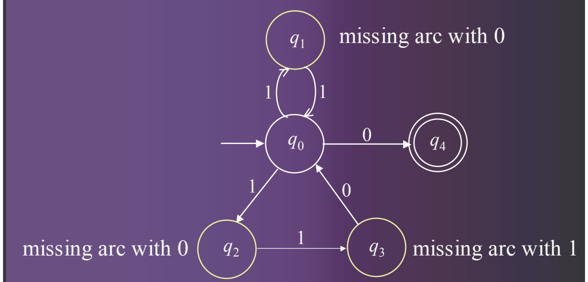
- But what about this, for accepting: **{11, 110}*{0}**




- We can read the expression in the diagram directly!



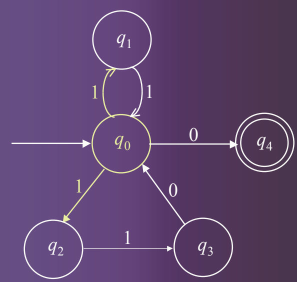
- But what about this, for accepting: **{11,110}*{0}**




- But we have states with no transitions on all symbols of Σ: No problem: we assume there is a non-accepting *sink state*



- But what about this, for accepting: **{11,110}*{0}**




- A state has two different next states on the same symbol!
- This FA is not deterministic: NFA

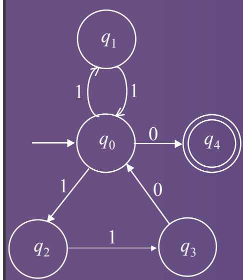


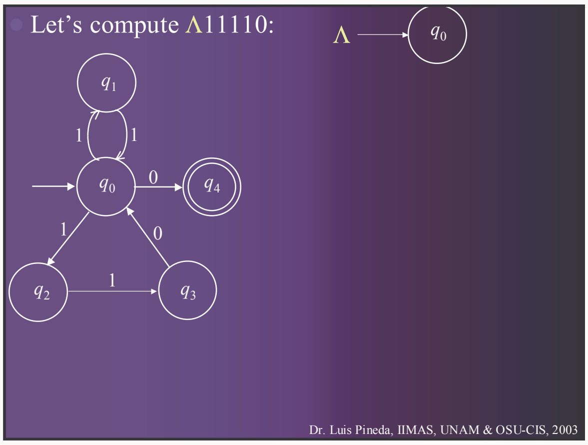
## Concept of non-determinism

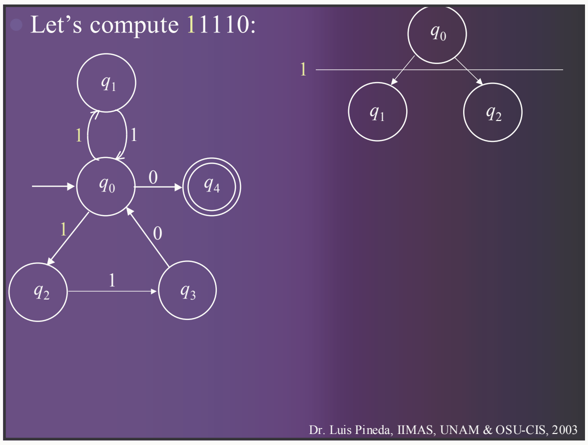
- An FA is not deterministic if there is more than one next state for any state on the same input symbol
- We can think of it in several ways:
  - As *n* DFAs running in parallel, each one taking care of a given path
  - As a automata that "guesses" the next state when it has a choice
  - As an abstract specification of a computation, regardless the actual details of the algorithm or machine that performs the computation
  - Non-determinism allows to think disjunctively about FA!

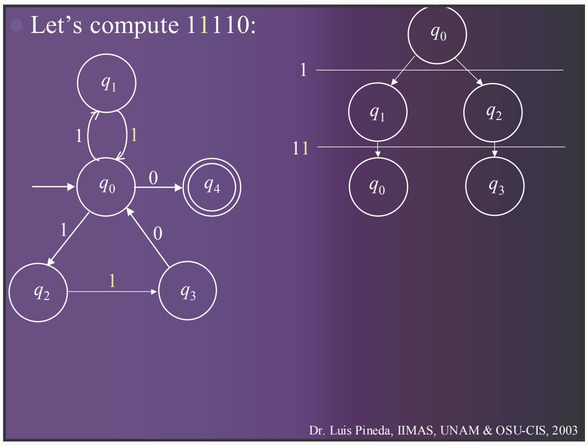


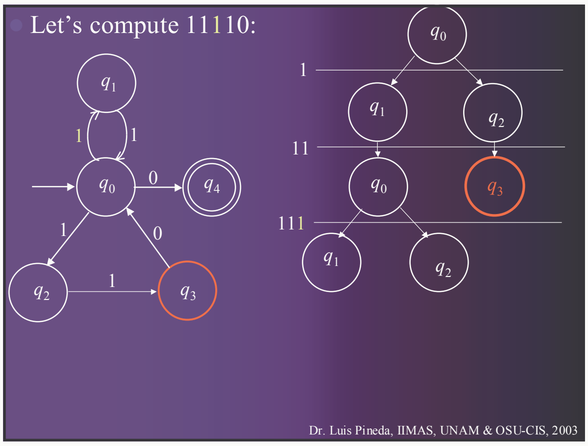
- Let's compute 11110:

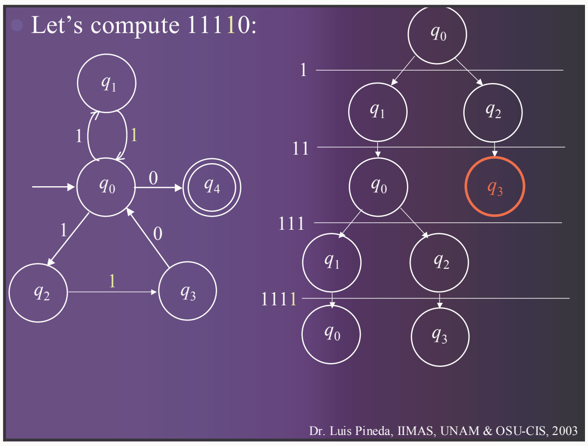


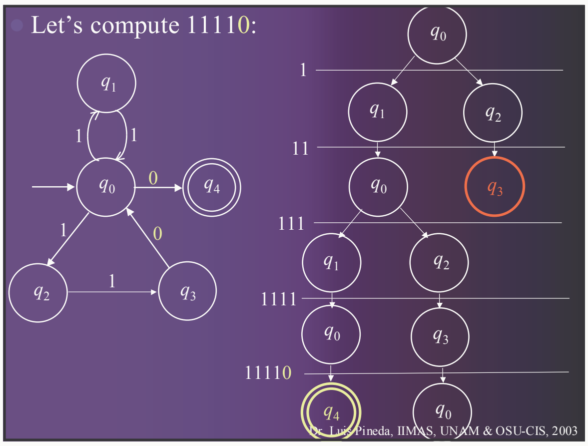

Let's compute Λ11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

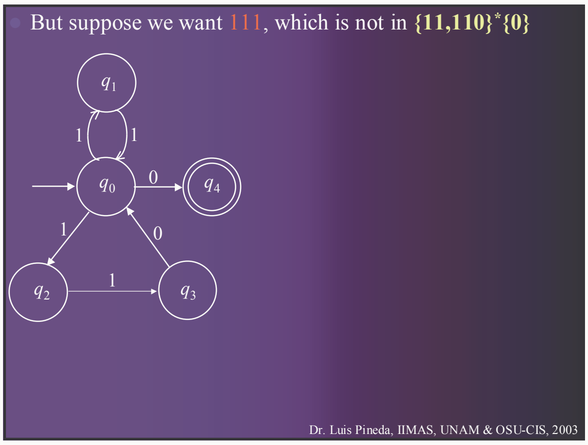

Let's compute 11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

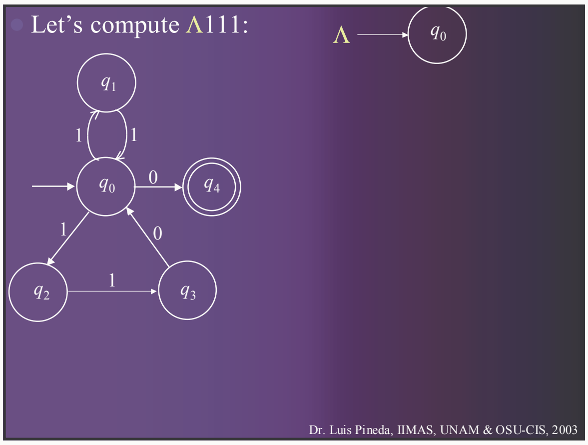

Let's compute 11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

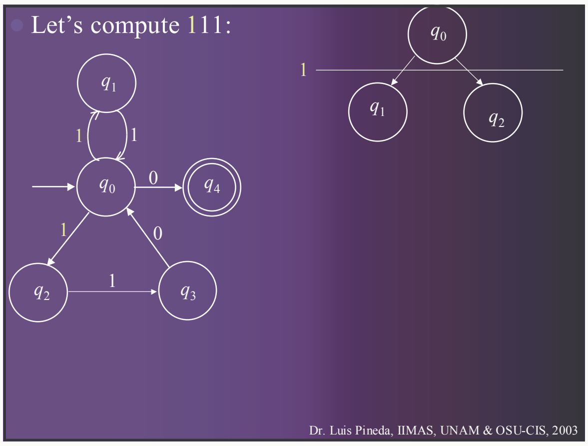

Let's compute 11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

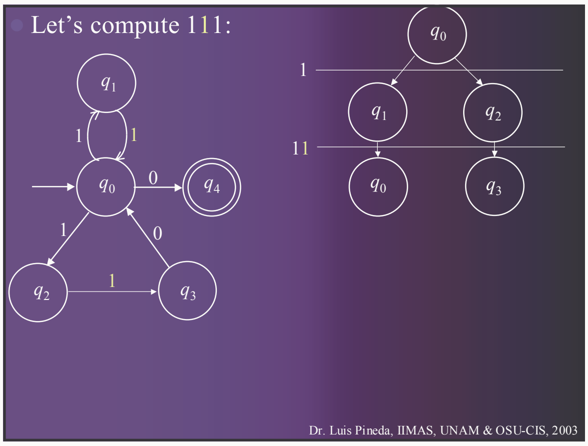

Let's compute 11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

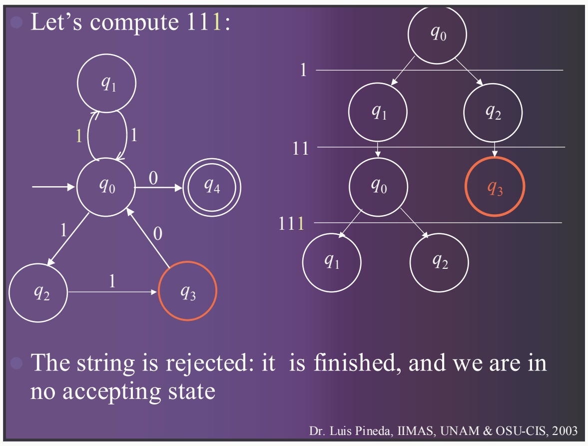

Let's compute 11110:
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

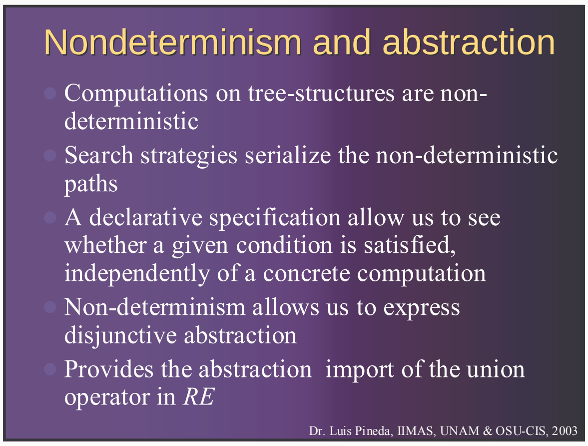

But suppose we want 111, which is not in {11,110}*{0}
q1
1
1
q0
0
q4
1
0
q2
1
q3
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


Let's compute Λ111:
Λ
q0
q1
1
1
q0
0
q4
1
0
q2
1
q3
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


Let's compute 111:
q0
1
q1
q2
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


Let's compute 111:
q0
1
q1
q2
11
q0
q3
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


Let's compute 111:
q0
1
q1
q2
11
q0
q3
111
q1
q2
The string is rejected: it is finished, and we are in no accepting state
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


Nondeterminism and abstraction
Computations on tree-structures are non-deterministic
Search strategies serialize the non-deterministic paths
A declarative specification allow us to see whether a given condition is satisfied, independently of a concrete computation
Non-determinism allows us to express disjunctive abstraction
Provides the abstraction import of the union operator in RE
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

## Definition of NFA

- A non-deterministic *finite Automaton* (NFA) is a 5-tuple $M = (Q, \Sigma, q_0, A, \delta)$, where
  - $Q$ is a finite set (of states)
  - $\Sigma$ is a finite alphabet
  - $q_0 \in Q$ (the initial state)
  - $A \subseteq Q$ (the set of accepting states)
  - $A$ transition function:

    $\delta: Q \times \Sigma \rightarrow 2^Q$
- The only difference between DAF y NFA is the type $\delta$



## Transition function

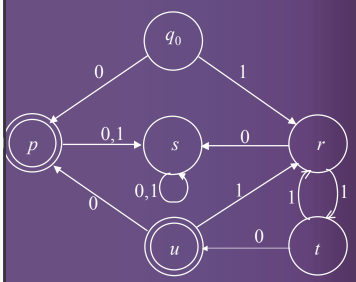
- Transition function for DFA:

  $\delta: Q \times \Sigma \rightarrow Q$
- Transition function for NFA:

  $\delta: Q \times \Sigma \rightarrow 2^Q$
- The type of the range of $\delta$ is a set of states!



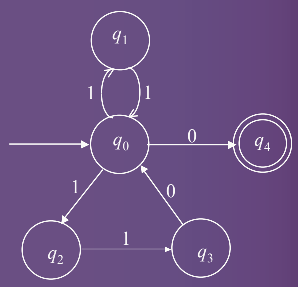
## Transition Table DFA




| | 0 | 1 |
|---|---|---|
| $q_0$ | $p$ | $r$ |
| $p$ | $s$ | $s$ |
| $s$ | $s$ | $s$ |
| $r$ | $s$ | $t$ |
| $t$ | $u$ | $r$ |
| $u$ | $p$ | $r$ |



## Transition Table NFA




| | 0 | 1 |
|---|---|---|
| $\rightarrow q_0$ | $\{q_4\}$ | $\{q_1, q_2\}$ |
| $q_1$ | $\phi$ | $\{q_0\}$ |
| $q_2$ | $\phi$ | $\{q_3\}$ |
| $q_3$ | $\{q_0\}$ | $\phi$ |
| $^*q_4$ | $\phi$ | $\phi$ |



## Extended Transition function for NFA

- Let $M = (Q, \Sigma, q_0, A, \delta)$ be a NFA.
- The function

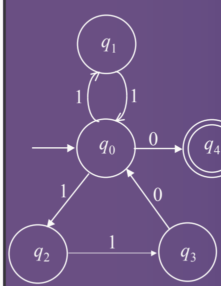
  $\delta^*: Q \times \Sigma^* \rightarrow 2^Q$

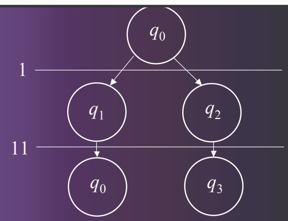
  is defined as follows:
  - For any $q \in Q$, $\delta^*(q, \Lambda) = \{q\}$
  - For any $q \in Q$, $y \in \Sigma^*$ and $a \in \Sigma$:

$$\delta^*(q, ya) = \bigcup_{r \in \delta^*(q,y)} \delta(r, a)$$



- Intuitively $\delta^*(q_0, 11)$ is:



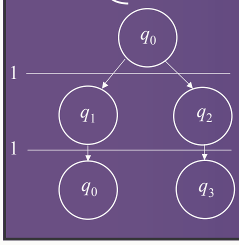




$\delta^*(q_0, 11) = \{q_0, q_3\}$

- Formally, $\delta^*(q_0, 11)$ is:

$\delta^*(q_0, 11)$ has the form $\delta^*(q_0, ya)$

Def. of $\delta^*$:
$$\delta^*(q, \Lambda) = \{q\}$$
$$\delta^*(q, ya) = \bigcup_{r \in \delta^*(q,y)} \delta(r, a)$$




$r \in \delta^*(q_0, \Lambda) = \{q_0\}$

$\delta^*(q_0, \Lambda 1) = \delta(q_0, 1) = \{q_1, q_2\}$

$r \in \delta^*(q_0, \Lambda 1) = \{q_1, q_2\}$

$\delta^*(q_0, \Lambda 11) = \delta(q_1, 1) \cup \delta(q_2, 1) = \{q_0, q_3\}$



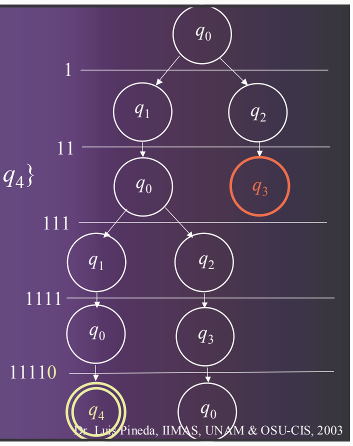
## Acceptance by an NFA

- Let $M = (Q, \Sigma, q_0, A, \delta)$ be a NFA.
  - The string $x \in \Sigma^*$ is accepted by $M$ if
    $\delta^*(q_0, x) \cap A \neq \emptyset$
  - The language recognized by $M$ is the set $\mathbf{L}(M)$ of all strings accepted by $M$
  - For any language $L \subseteq \Sigma^*$, $L$ is recognized by $M$ if $L = \mathbf{L}(M)$



- Accepting 11110:

$\delta^*(q_0, \Lambda 11110) = \{q_0, q_4\}$



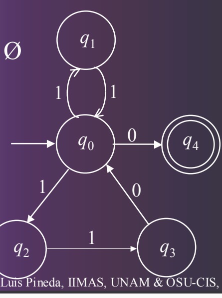



- Accepting 11110:

$\delta^*(q_0, \Lambda 11110) = \{q_0, q_4\}$

$A = \{q_4\}$

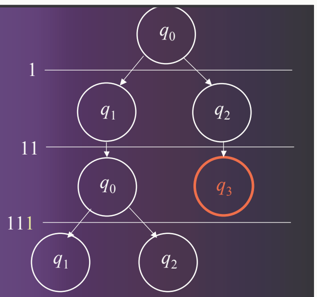
$\delta^*(q_0, \Lambda 11110) \cap A = \{q_4\} \neq \emptyset$






- But if the string is 111:

$\delta^*(q_0, \Lambda 111) = \{q_1, q_2\}$






- Accepting 111:

$\delta^*(q_0, \Lambda 111) = \{q_1, q_2\}$

$A = \{q_4\}$

$\delta^*(q_0, \Lambda 111) \cap A = \emptyset$