DAY 4 Theory of Computation

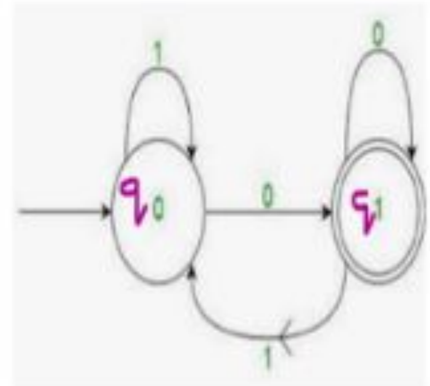
DAY 4 CLASS (23/09/2020, 9AM -10 AM)

1. Types of Finite Automaton
2. DFA (Deterministic finite Automaton)
3. NFA (Non Deterministic Finite Automaton)
4. Acceptability by NFA.

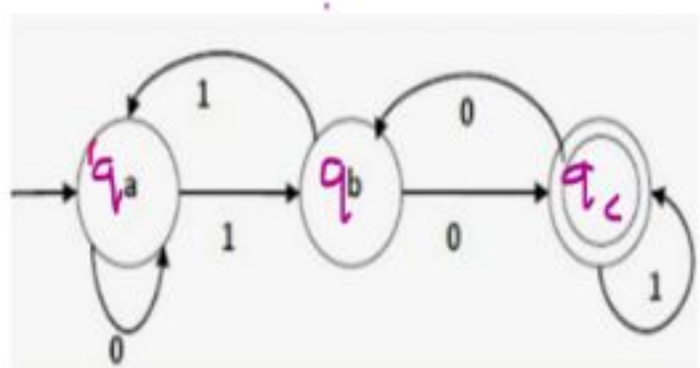
png12

Types of FA(Finite Automaton)

Example of DFA



1.DFA (Deterministic FA)
2.NFA (Non Deterministic FA)



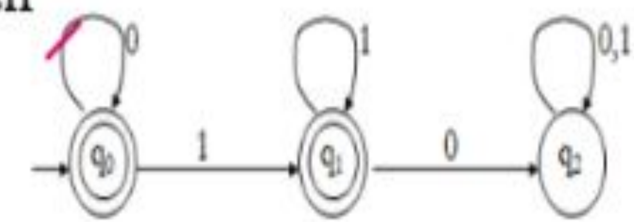
If every state is applied with all possible inputs in TD, DFA

DFA(Deterministic Finite Automaton

Wednesday, September 23, 2020 9:30 AM

Example

All symbols applied to all states, DFA



DFA Example

alphabet $\Sigma = \{0, 1\}$
start state $Q = \{q_0, q_1, q_2\}$
initial state q_0
accepting states $F = \{q_0, q_1\}$

transition function δ:

inputs

states	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

filled

cell

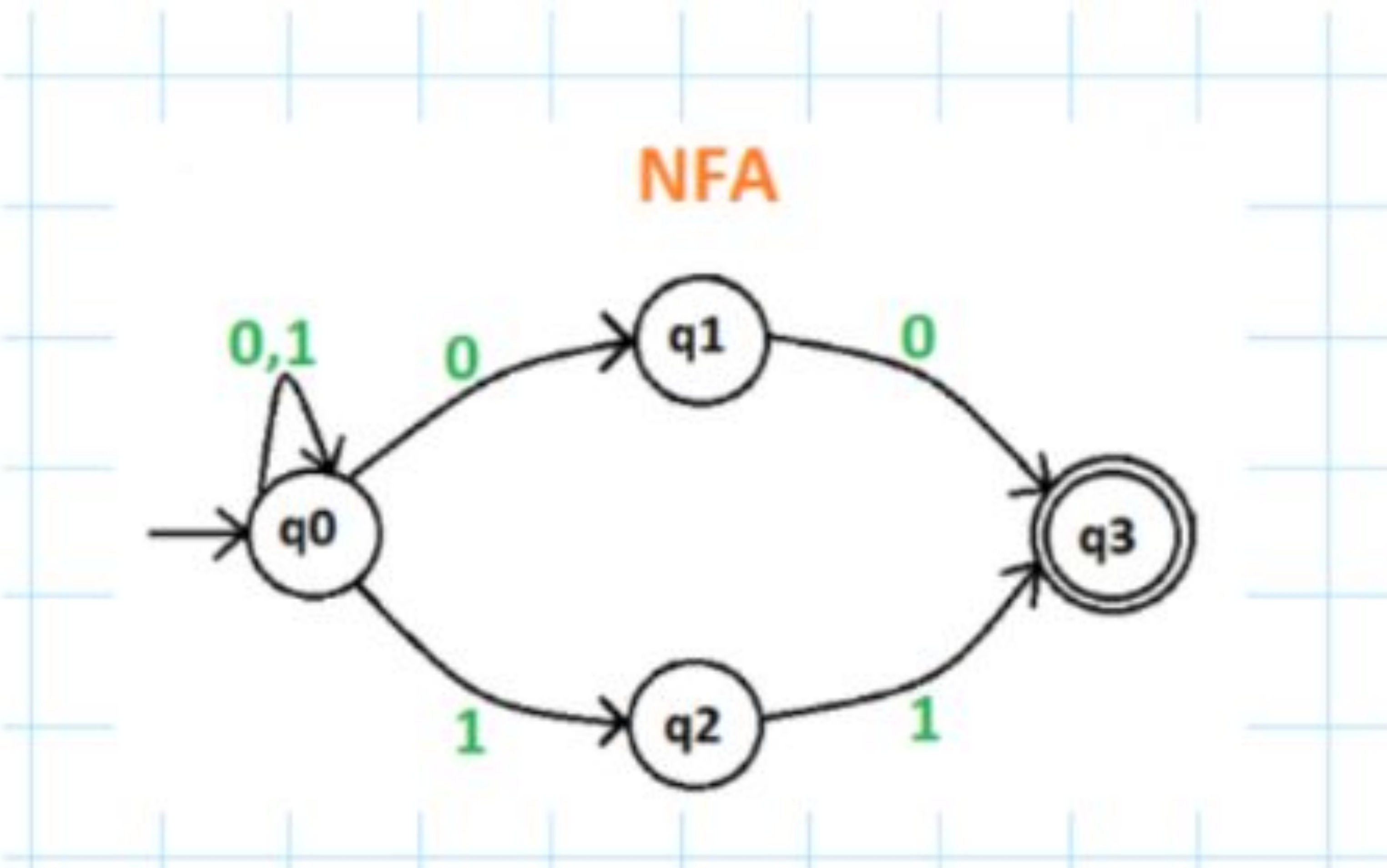
Non Deterministic Finite Automata (NFA):

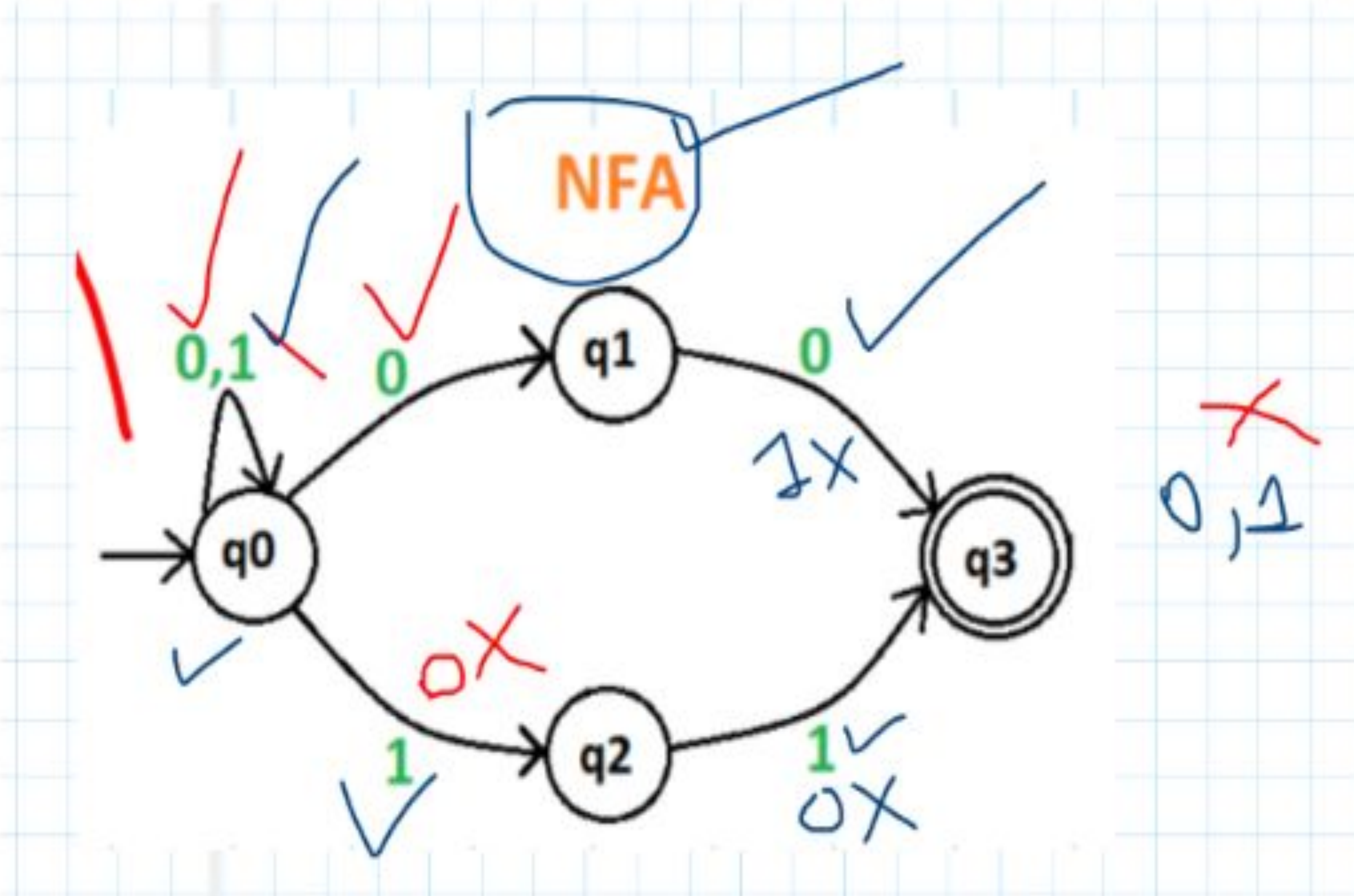
Like the DFA, a NFA has a finite set of states, a finite set of input symbols, one start state, a set of accepting states and transition function (δ).

The difference between the DFA and the NFA is in the type of δ. For the NFA, δ is a function that takes a state and input symbol as arguments as like the DFA's transition function, but returns a set of **zero, one or more state (DFA returns exactly one state).**

So, we can conclude in a NFA, there **may be zero no. of arcs** out of each state for each input symbol. While in DFA, it **has exactly one arc** out of each state for each input symbol.

Example of NFA




NFA
0,1
0
q0
q1
0
q2
1
1
q3
0,1

Nondeterministic Finite Automaton

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

where

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

$F = \{q_2, q_4\}$

δ	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\varnothing$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	$\varnothing$
q_4	$\{q_4\}$	$\{q_4\}$

More Than One entry

one entry

No entry

pe Text He

Non-deterministic Finite Automata (NFA)

- A Non-deterministic Finite Automaton (NFA) consists of:
  - Q ==> a finite set of states
  - ∑ ==> a finite set of input symbols (alphabet)
  - q_0 ==> a start state
  - F ==> set of accepting states
  - δ ==> a transition function, which is a mapping between Q x ∑ ==> subset of Q
- An NFA is also defined by the 5-tuple:
  - {Q, ∑ , q_0,F, δ }

Why Subset of Q

as incase of NFA there may be zero, one or more number next states for same input symbol and

Let Q={q0,q1,q2}
Subset of Q={null,{q0},{q1},{q2},{q0,q1},{q1,q2},{qo,q2},{qo,q1,q2}}

1. No next state: null
2. One next state: q0 / q1 / q2
3. More than one next state for same input symbol like qo and q1, q1 and q2 , qo and q1 and q2

Non Deterministic Finite Automata (NFA):

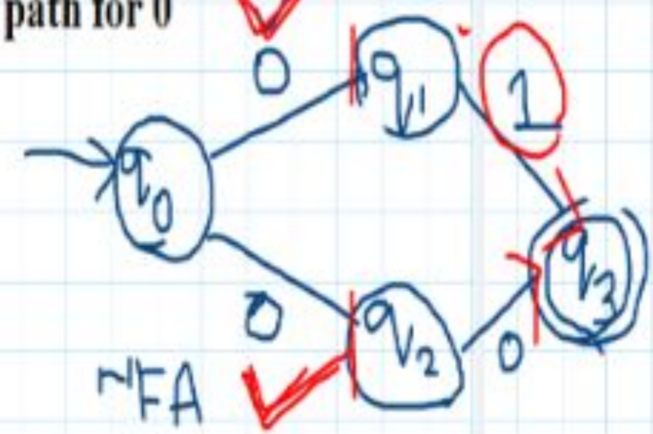
Like the DFA, a NFA has a finite set of states, a finite set of input symbols, one start state, a set of accepting states and transition function (δ).

The difference between the DFA and the NFA is in the type of δ. For the NFA, δ is a function that takes a state and input symbol as arguments as like the DFA's transition function, but returns a set of **zero, one or more state (DFA returns exactly one state).**

So, we can conclude in a NFA, there **may be zero no. of arcs** out of each state for each input symbol. While in DFA, it **has exactly one arc** out of each state for each input symbol.

NFA, Multiple path for same input

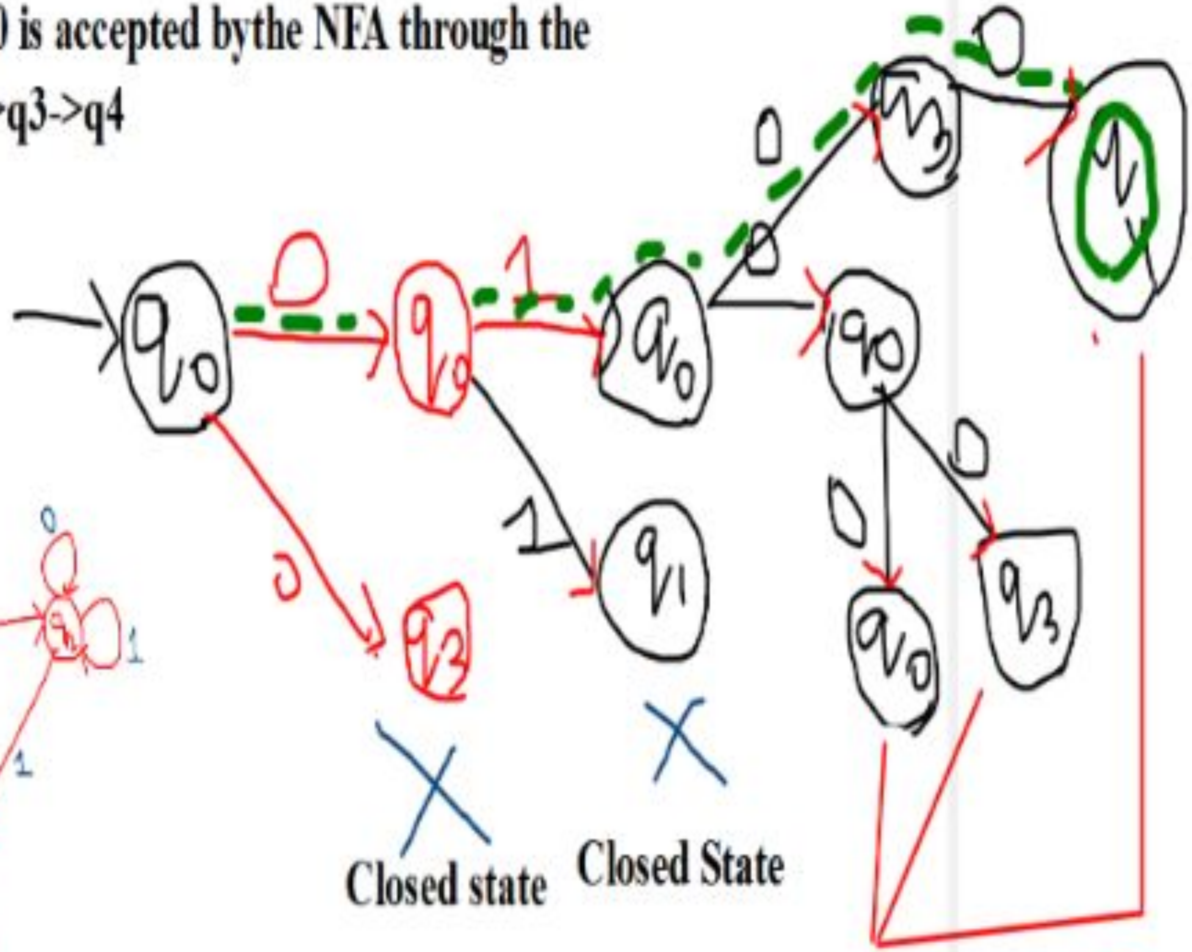
1.One path from q1 for 1 symbol, no path for 0

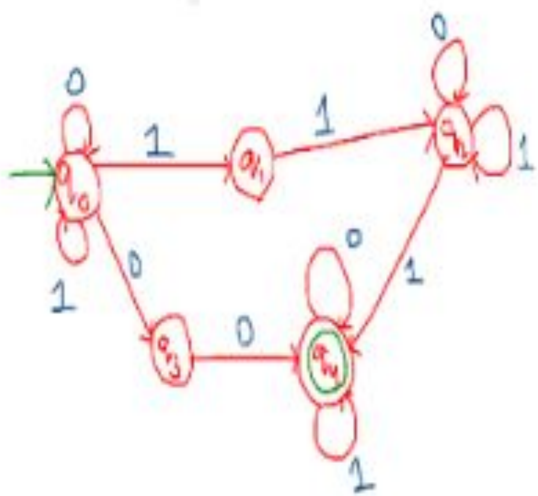


1.For state q0, same input 0 is applied twice , there are more than one next state (q1, q2)

3. no next state from previous state q3

The string w= 0110 is accepted bythe NFA through the path q0->q0->q0->q3->q4





Closed state

Closed State

Terminal States
qo,q3,q4
q4 is the final state