Automata and Language Theory

Non-Deterministic Finite Automata

Finite Automata (cont)

- **Nondeterministic Finite Automata (NFA)**
  - A type of finite state machine used in computational theory to recognize languages.
  - Unlike DFA, an NFA can have:
    - ✓ **Multiple transition for the same symbol from a single state.**
    - ✓ **ε-moves** (epsilon transitions), which allow the machine to transition between states without consuming input
    - ✓ Multiple possible paths to reach an accepting state

Finite Automata (cont)

- **Nondeterministic Finite Automata (NFA)**
  - An NFA can be represented by a 5-tuple (**Q**, **∑**, **δ**, q_0, **F**) where

✓ **Q** is finite set of states

✓ ∑ (sigma, summation) is an input alphabet

✓ δ (lowercase delta) is a transition function
$\delta: Q \times \Sigma\varepsilon \rightarrow P(Q)$ (can go to multiple states)

✓ q_0 A initial state from where any input is processed
($q_0 \in Q$: q_0 is one of the states of Q; ∈ - "is an element")

✓ F is a set of final state/states of Q ($F \subseteq Q$ -> set of accepting states is a subset of the total set of states).

✓ δ: Q × Σε→P(Q) (can go to multiple states)

Understanding Each Symbol

- Q → The set of states in the NFA.
- Σ_ε → The set of input symbols, including **ε (epsilon)**, which represents an empty transition.
  - Σ is the input alphabet.
  - $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ means that the input can be any symbol from Σ or an **epsilon transition**.
- $\times$ **(Cartesian Product)** → This means the function takes a pair **(state, input symbol)** as input.
- $P(Q)$ → The **power set** of Q, which represents **a set of states**.
  - Since an NFA can transition to **multiple states** for the same input, the output is a **set** of states.

Finite Automata (cont)

- **Nondeterministic Finite Automata (NFA)**

➢ STEPS

1. Define the NFA components $M=(Q,\Sigma,\delta,q0,F)$
2. Start at the initial state
   * begins at the initial state q_0 ;
   * if there are ε-–transitions, it can move to new states without consuming input
3. Read the First input symbol
   * check the current set of states
   * look at the transition function for the current states
   * The NFA can move to multiple states for the same input symbol

○ **Nondeterministic Finite Automata (NFA)**

➢ STEPS

4. Move to Next Possible States

    * If there are multiple transitions, follow all possible paths at once
    * If ε—transitions exist, move before processing the next input

5. Continue until the String is processed

    * Keep following transitions until all input symbols are read ;
    * Keep track of all active states at each step

6. Check if any state is accepting

    * If any of the current states at the end of the input belongs to F (final states, the string is accepted, otherwise, the string is rejected.

1. Construct a DFA and the NFA of the set of all strings that starts with 0.

L = {0, 01, 00, 001}   1, 100, 111

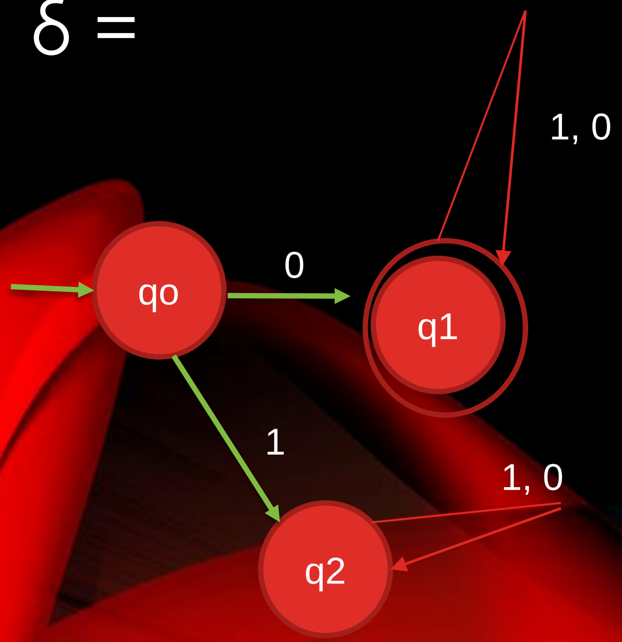
DFA

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

Q = qo, q1, q2

F = q1

δ =



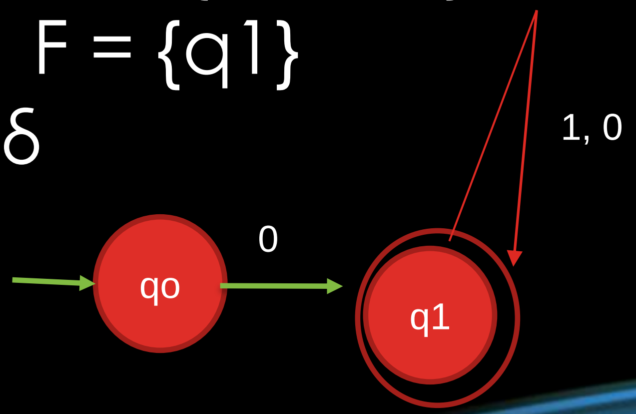
NFA

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

Q = {qo, q1}

F = {q1}

δ



qo = qo, q1, qoq1, ε

2. Construct the DFA and NFA that accepts sets of all strings over {0,1} of length 2.

L = { 00, 01, 10, 11 }   {0, 1, 000, 111, 00000}

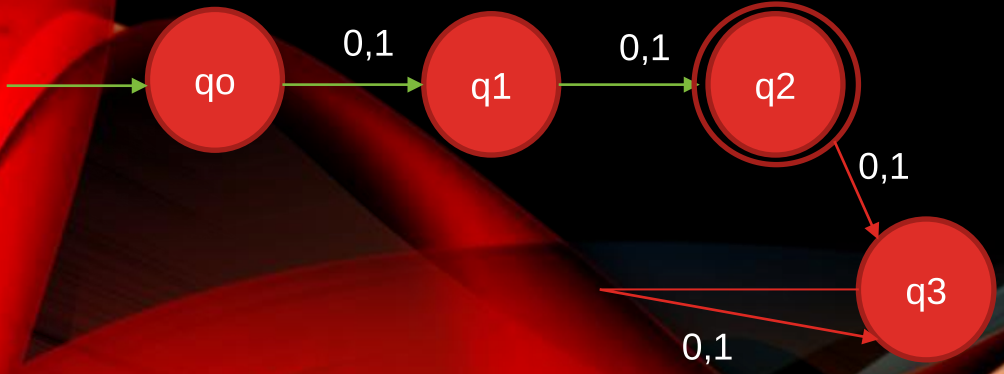
DFA

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

Q = {q0, q1, q2, q3}

F = {q2}

δ =

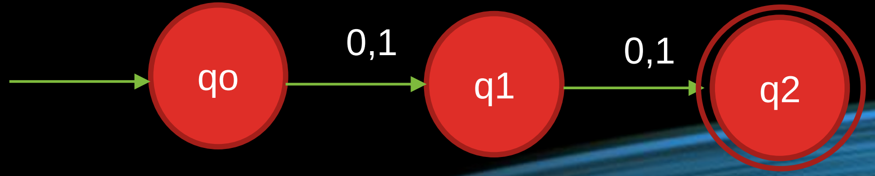


NFA

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

Q = {qo, q1, q2}

F = {q2}



qo = qo, q1, q2, q1q2, q0q1
q0q2, qoq1q2, ε

3. Construct the DFA and NFA that accepts all strings that ends with 1.

4. Construct the DFA and NFA where all set of strings contains 0.

5. Construct the DFA and NFA where set of strings starts with 10.