**CS304 - Formal Languages and Automata Theory** $4^{th}$ **Semester**

# Lecture 3 : Non-Deterministic Finite Automata

*Instructor: Prof. Prateek Vishnoi* *Indian Institute of Technology, Mandi*

We now try to extend the power of DFAs by allowing multiple possible transitions for a given state and input symbol. This model of computation is known as a Nondeterministic Finite Automaton (NFA).

## 3.1 Nondeterministic Finite Automata (NFA)

A nondeterministic finite automaton (NFA) is a finite automaton where, for a given state and input symbol, the machine may move to **multiple possible next states** (including no state).

An input is accepted if **at least one computation path** leads to an accepting state.

### 3.1.1 Formal Definition

An NFA is a 5-tuple

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

where

- $Q$ : finite set of states
- $\Sigma$ : input alphabet
- $\delta : Q \times \Sigma \to 2^Q$ (transition function)
- $q_0 \in Q$ : start state
- $F \subseteq Q$ : set of accepting states

### 3.1.2 Key Difference from DFA

$$\delta_{\text{DFA}} : Q \times \Sigma \to Q$$

$$\delta_{\text{NFA}} : Q \times \Sigma \to 2^Q$$

Thus an NFA may:

- have multiple next states
- have zero next states
- follow many paths simultaneously

### 3.1.3 Extended Transition Function

We extend $\delta$ to strings.

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

**Basis**

$$\delta^*(q, \epsilon) = \{q\}$$

**Induction** For $w \in \Sigma^*$ and $a \in \Sigma$,

$$\delta^*(q, wa) = \bigcup_{p \in \delta^*(q,w)} \delta(p, a)$$

### 3.1.4 Language Accepted by an NFA

An NFA accepts a string $w$ if

$$\delta^*(q_0, w) \cap F \neq \emptyset$$

Thus,

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$$

### 3.1.5 Example of NFA

Consider an NFA over $\Sigma = \{0, 1\}$ that accepts strings ending in 01.

$$Q = \{q_0, q_1, q_2\}, \quad q_0 \text{ start}, \quad F = \{q_2\}$$

Transition table:

| | 0 | 1 |
|---|---|---|
| $q_0$ | $\{q_0, q_1\}$ | $\{q_0\}$ |
| $q_1$ | $\emptyset$ | $\{q_2\}$ |
| $q_2$ | $\emptyset$ | $\emptyset$ |

Explanation:

- From $q_0$ on input 0, machine guesses start of suffix 01.
- From $q_1$ on input 1, it reaches accepting state $q_2$.

Next relevant question we can have is :

*Is NFA more powerful than DFA??*

Precisely, can NFA accept more languages than DFA. We will discover this in the next lecture.

## References

[1] Michael Sipser, *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning, 2012.