Non Deterministic Finite Automata
&
Deterministic Finite Automata

Course Name: Compiler Design
Course Code: CSE331
Level:3, Term:3
Department of Computer Science and Engineering
Daffodil International University

Introduction

The term "Automata" is derived from the Greek word "αὐτόματα" which means "self-acting". An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically.

An automaton with a finite number of states is called a Finite Automaton (FA) or Finite State Machine (FSM).

An automaton can be represented by a 5-tuple (Q, ∑, δ, q0, F), where −

- Q is a finite set of states.
- ∑ is a finite set of symbols, called the alphabet of the automaton.
- δ is the transition function.
- q0 is the initial state from where any input is processed ($q0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

Related Terminologies

- **Alphabet**

Definition: An alphabet is any finite set of symbols.

Example: ∑ = {a, b, c, d} is an alphabet set where ‘a’, ‘b’, ‘c’, and ‘d’ are symbols.

- **String**

Definition: A string is a finite sequence of symbols taken from ∑.

Example: ‘cabcad’ is a valid string on the alphabet set ∑ = {a, b, c, d}

- **Length of a String**

Definition: It is the number of symbols present in a string. (Denoted by |S|).

Examples: If S = ‘cabcad’, |S|= 6

If |S|= 0, it is called an empty string (Denoted by λ or ε)

- **Kleene Star**

Definition: The Kleene star, ∑*, is a unary operator on a set of symbols or strings, ∑, that gives the infinite set of all possible strings of all possible lengths over ∑ including λ.

Representation: ∑* = ∑0 ∪ ∑1 ∪ ∑2 ∪……. where ∑p is the set of all possible strings of length p.

Example: If ∑ = {a, b}, ∑* = {λ, a, b, aa, ab, ba, bb,………..}

Related Terminologies

- **Kleene Closure / Plus**

Definition: The set ∑+ is the infinite set of all possible strings of all possible lengths over ∑ excluding λ.

Representation: $\Sigma^+ = \Sigma 1 \cup \Sigma 2 \cup \Sigma 3 \cup \ldots\ldots$

$\Sigma^+ = \Sigma^* - \{ \lambda \}$

Example: If ∑ = { a, b } ,
$\Sigma^+ = \{ a, b, aa, ab, ba, bb, \ldots\ldots\ldots\}$

- **Language**

Definition: A language is a subset of ∑* for some alphabet ∑. It can be finite or infinite.

Example: If the language takes all possible strings of length 2 over ∑ = {a, b}, then L = { ab, aa, ba, bb }

Types of Finite Automaton

Finite Automaton can be classified into two types –

- Deterministic Finite Automaton (DFA)
- Non-deterministic Finite Automaton (NDFA / NFA)

Non Deterministic Finite Automaton (NFA)

A NFA (Nondeterministic Finite Automata) is a mathematical model that consists of 5 tuples

- A set of states S
- A set of input symbols Σ (the input symbol alphabet)
- A transition function move that maps state-symbol pairs to sets of states
- A state So that is distinguished as the start (or initial) state.
- A set of states F distinguished as accepting (or final) states.

Formal Definition of a DFA

A DFA can be represented by a 5-tuple (Q, ∑, δ, q0, F) where −

- Q is a finite set of states.
- ∑ is a finite set of symbols called the alphabet.
- δ is the transition function where $\delta: Q \times \sum \rightarrow Q$
- q0 is the initial state from where any input is processed ($q0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

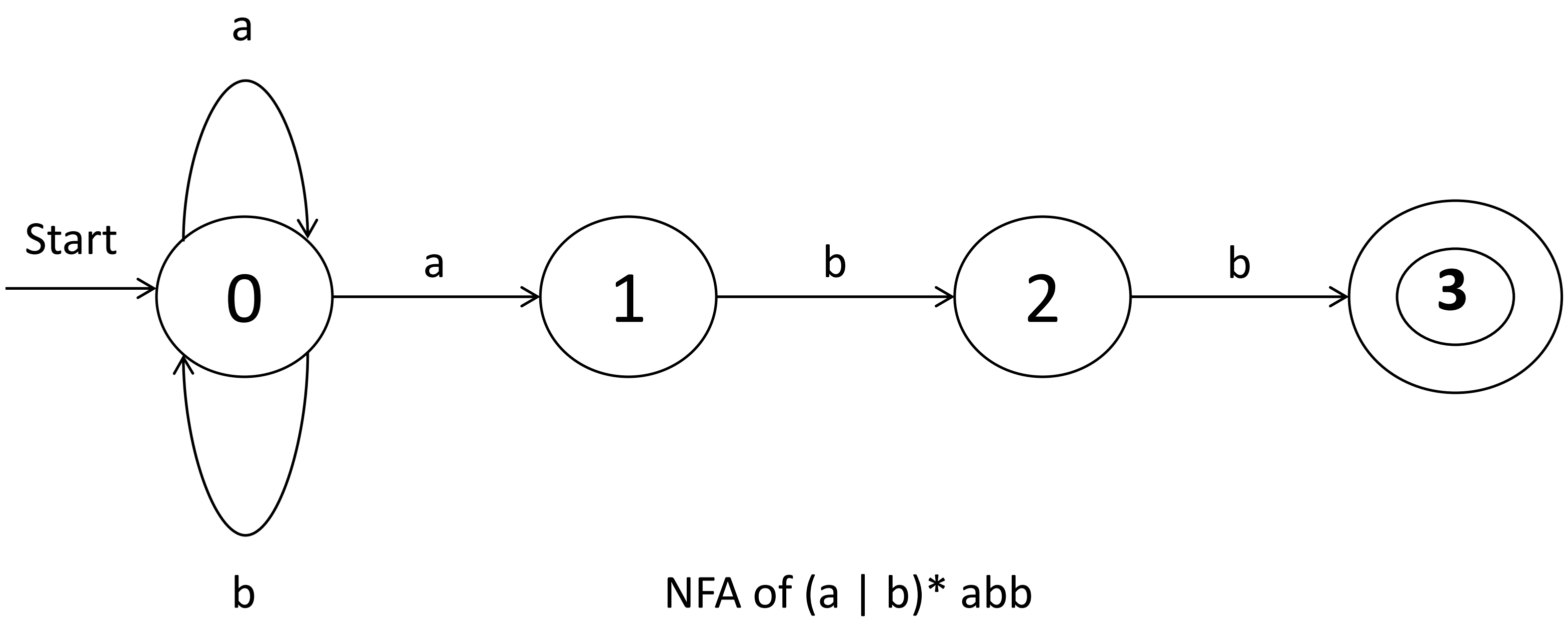
Graphical Representation of a NFA

An NFA is represented by digraphs called state diagram.

- The vertices represent the states.
- The arcs labeled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

Example

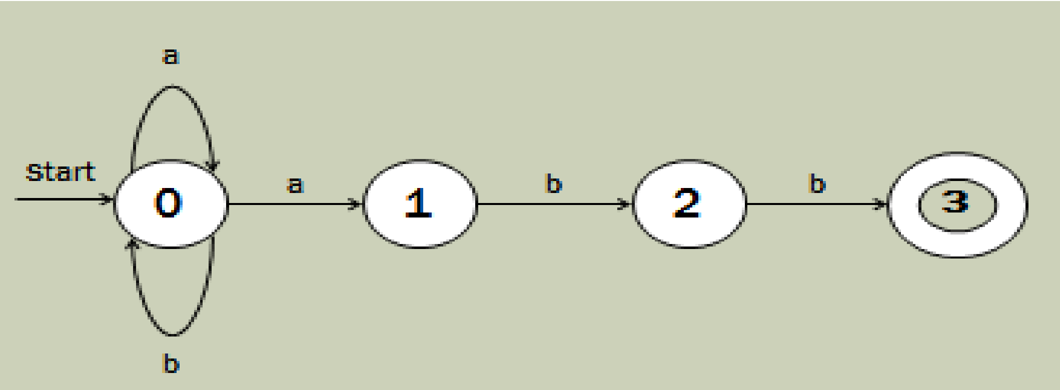
Regular Expression: (a | b)* abb



NFA of (a | b)* abb

Example

- Lets find the 5 tuples
- States: {0, 1, 2, 3}
- Input Symbol: {a, b}
- Start State: {0}
- Final State: {3}
- Transition function: Transition table shows the function



State	Input Symbol	
	a	b
0	{0, 1}	{0}
1	--	{2}
2	--	{3}

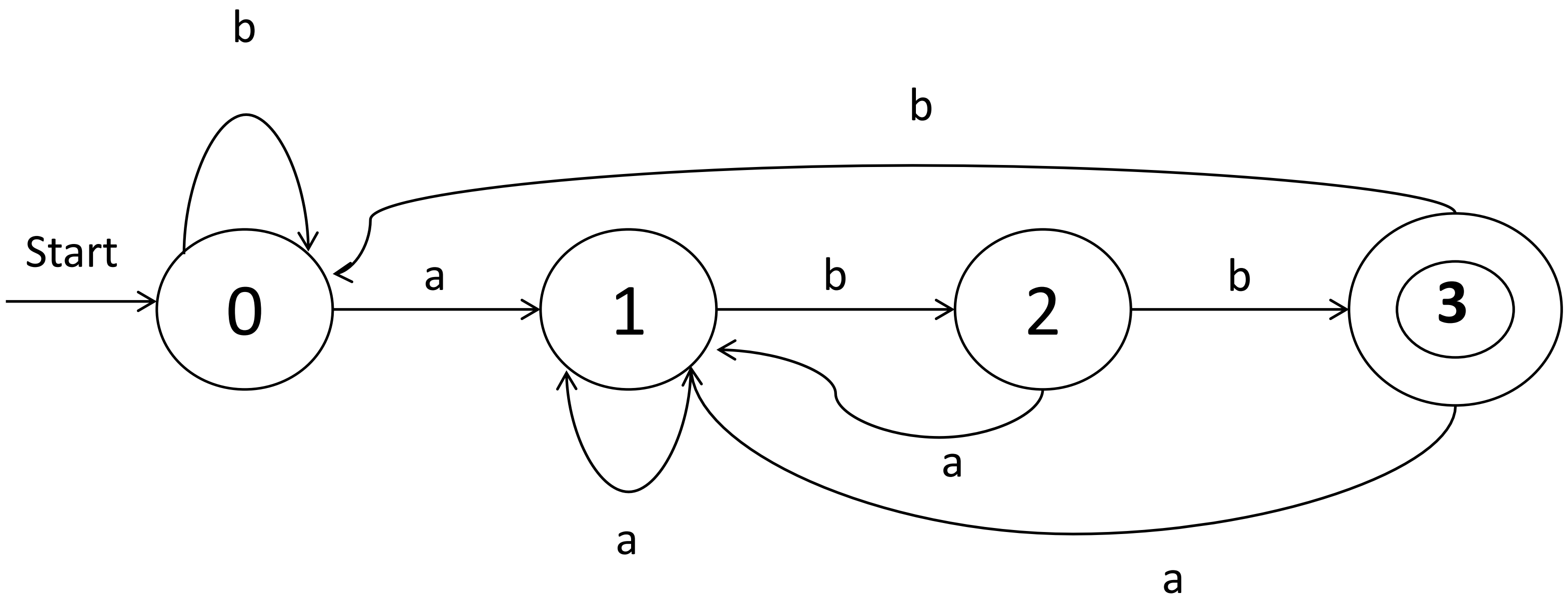
Graphical Representation of a DFA

A DFA is represented by digraphs called state diagram.

- The vertices represent the states.
- The arcs labeled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

Example

- Regular Expression: (a | b)* abb



Example

Let a deterministic finite automaton be - $Q = \{a, b, c\}$,

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

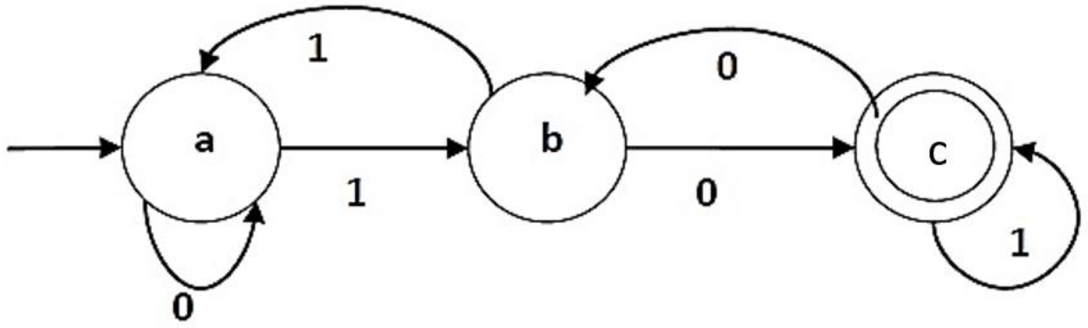
$q0 = \{a\}$,

$F = \{c\}$,

Transition function δ as shown by the following table -

Present State	Next State for Input 0	Next State for Input 1
a	a	b
b	c	a
c	b	c

Its graphical representation would be as follows -



DFA vs NFA

DFA	NFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called deterministic.	The transition from a state can be to multiple next states for each input symbol. Hence it is called non-deterministic.
Empty string transitions are not seen in DFA.	NFA permits empty string transitions.
Backtracking is allowed in DFA	In NFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NFA, if at least one of all possible transitions ends in a final state.

THANK YOU