Welcome To My Presentation

My Presentation Topics is

“DFA for the language of all those strings starting and ending with different letters”

Presented By

Tushar Sarkar

Student ID : 18CSE035

Second Year Second Semester

Department of CSE, BSMRSTU.

Content

- Symbol, Alphabet
- Strings, Languages
- What is DFA?
- Formal Definition of DFA
- How to use an DFA?
- Problem Solve
- Final Result
- Application of DFA

Symbol, Alphabet

- Symbol :

  Symbol is the basic building block of Theory of computation.
  Example :

  a, b, c, d, ……, z
  0, 1, 2, 3, 4, 5, …, 9
  letter, digits, etc.

- Alphabet

  An alphabet is a finite, non-empty set of symbols which denote by ∑ (sigma).
  Examples :

  Binary: ∑ = {0,1}
  All lower case letters: ∑ = {a,b,c,..z}.
  Alphanumeric: ∑ = {a-z, A-Z, 0-9}

Strings, Languages

- Strings :
  - A string or word is a finite sequence of symbols chosen from ∑
  - Length of a string w, denoted by "|w|", is equal to the number of (non- ε) characters in the string.

Example: w=0001,1010,101010.

- Languages :

Languages is the collections of strings which can be finite and infinite.

- *L is a said to be a language over alphabet ∑, only if* $L \subseteq \sum^*$
- this is because $\sum^*$ is the set of all strings (of all possible length including 0) over the given alphabet ∑

Examples:

Let L be *the* language of all strings consisting of *n* 0's followed by *n* 1's:

L = {ε, 01, 0011, 000111,…}

Let L be *the* language of all strings of with equal number of 0's and 1's:

L = {ε, 01, 10, 0011, 1100, 0101, 1010, 1001,…}

What is DFA?

- DFA refers to **Deterministic Finite Automata**. Deterministic refers to the uniqueness of the computation. The finite automata are called finite automata if the machine is read an input string one symbol at a time.
- In DFA, there is only one path for specific input from the current state to the next state.
- DFA does not accept the null move, i.e., the DFA cannot change state without any input character.
- DFA can contain multiple final states. It is used in Lexical Analysis in Compiler.

Formal Definition of DFA

- A DFA is defined by the 5-tuple:

  $\{Q, \sum, q_0, F, \delta\}$

- Here,
  - Q: finite set of states
  - $\sum$: finite set of the input symbol
  - q_0: initial state
  - F: final state
  - δ: Transition function

How to use an DFA?

- Input: a word w in $\sum^*$
- Question: Is w acceptable by the DFA?
- Steps:
  - Start at the "start state" q_0
  - For every input symbol in the sequence w do
    - Compute the next state from the current state, given the current input symbol in w and the transition function
  - If after all symbols in w are consumed, the current state is one of the accepting states (F) then *accept w;*
  - Otherwise, *reject w.*

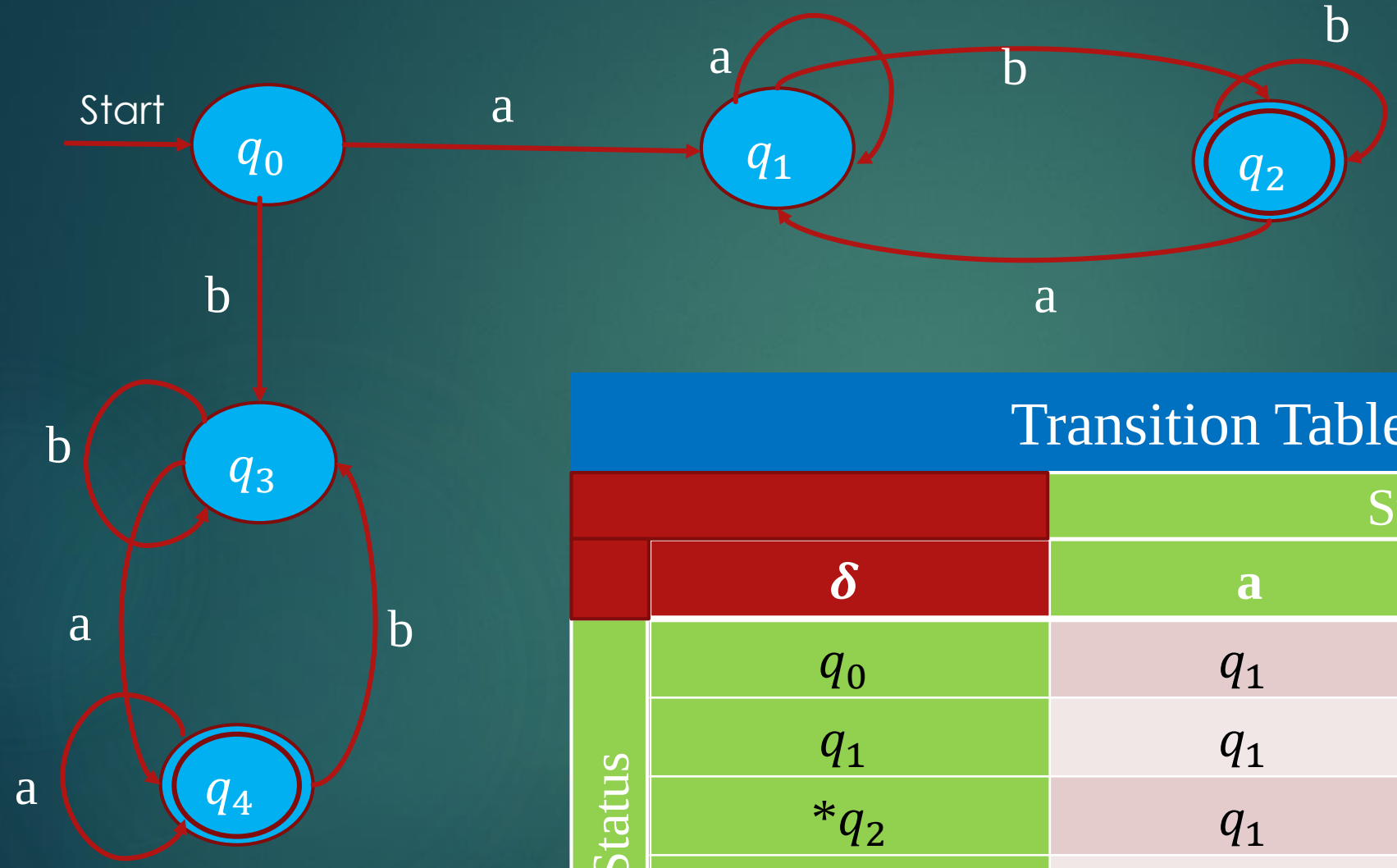
Problem Solve

- Question :

  DFA for the language of all those strings starting and ending with different letters

- Solve :
  - A finite set of states, Q = $\{q_0, q_1, q_2, q_3, q_4\}$
  - A finite set of input symbols, $\sum = \{a, b\}$
  - Start state = q_0
  - Set of accepting states, F = $\{q_2, q_4\}$
  - So Language, L={∈, ab, ba, ……}

Problem Solve



Transition Table

	δ	Symbol a	Symbol b
Status	q_0	q_1	q_3
	q_1	q_1	q_2
	$*q_2$	q_1	q_2
	q_3	q_4	q_3
	$*q_4$	q_4	q_3

Final Result

- So the DFA “Five-tuple” notion is

$$A = (\{q_0, q_1, q_2, q_3, q_4\}, \{a,b\}, \delta, q_0, \{q_2, q_4\})$$

Application of DFA

- DFA uses include **protocol analysis, text parsing, video game character behavior, security analysis**, CPU control units, natural language processing, and speech recognition.

Thank You