Welcome To My Presentation

My Presentation Topic is
Design an NFA in which all the string contain a substring 1100

Presented By

Tushar Sarkar
Student ID : 18CSE035
Second Year Second Semester
Department of CSE, BSMRSTU.

Outline

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

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 $\sum$ (sigma).
  Examples :

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

Strings, Languages

- **Strings :**

  - A string or word is a finite sequence of symbols chosen from $\sum$
  - 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 $\sum$, 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 $\sum$

Examples:

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

$L = \{\varepsilon, 01, 0011, 000111, \ldots\}$

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

$L = \{\varepsilon, 01, 10, 0011, 1100, 0101, 1010, 1001, \ldots\}$

What is NFA?

- NFA stands for non-deterministic finite automata. It is easy to construct an NFA than DFA for a given regular language.
- The finite automata are called NFA when there exist many paths for specific input from the current state to the next state.
- Every NFA is not DFA, but each NFA can be translated into DFA.
- NFA is defined in the same way as DFA but with the following two exceptions, it contains multiple next states, and it contains ε transition.

Formal Definition of NFA

- A NFA 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 : a transition function, which is a mapping between Q x $\sum$ ==> subset of Q

How to use an NFA?

- Input: a word w in $\sum^*$
- Question: Is w acceptable by the NFA?
- Steps:
  - Start at the "start state" q_0
  - For every input symbol in the sequence w do
    - Determine all possible next states from all current states, 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 :

  Design an NFA in which all the string contain a substring 1100.

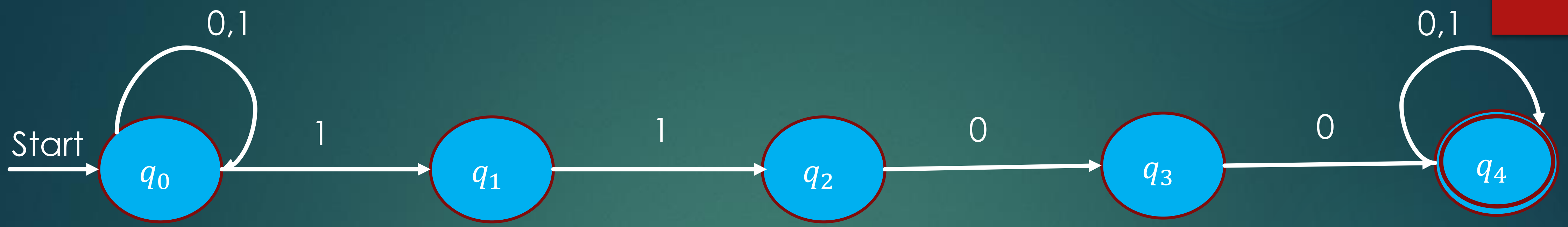
- Solve

  Here,

  L(M)={w|w which double ‘1’ is followed by double’0’}
  L = {1100, 0001100,11001100 ……..}.

  - A finite set of states, Q = $\{q_0, q_1, q_2, q_3, q_4\}$
  - A finite set of input symbols, $\sum = \{0,1\}$
  - Start state = q_0
  - Set of accepting states, F = $\{q_4\}$
  - δ is the transition function where δ: $Q \times \sum \rightarrow 2^Q$

Problem Solve



δ	Transition Table	
States	$Q \times \sum \rightarrow 2^Q$ 0	$Q \times \sum \rightarrow 2^Q$ **1**
q_0	q_0	q_0, q_1
q_1	Ø	q_2
q_2	q_3	Ø
q_3	q_4	Ø
$*q_4$	q_4	q_4

Application of NFA

- It is important because NFAs can be used to reduce the complexity of the mathematical work required to establish many important properties in the theory of computation.

- For example, it is much easier to prove closure properties of regular languages using NFAs than DFAs.

Thank you!