CSE211 – Formal Languages and Automata Theory

U1L7 – Deterministic Finite Automata

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

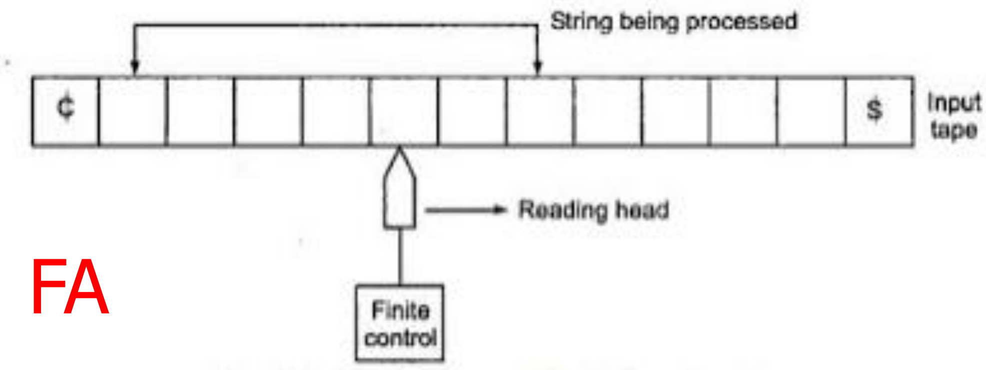
- Recap of previous class
- Types of Automata
- Pictorial representation of Automaton
- Introduction to Finite Automata
- Deterministic Finite Automata(DFA)
- Definition of DFA
- Notations of DFA
- Extended transition function of DFA
- Designing DFA-Examples

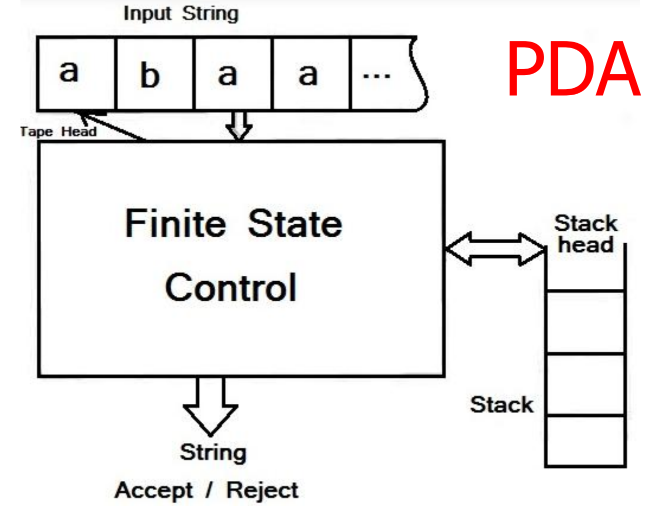
Kinds of Automata

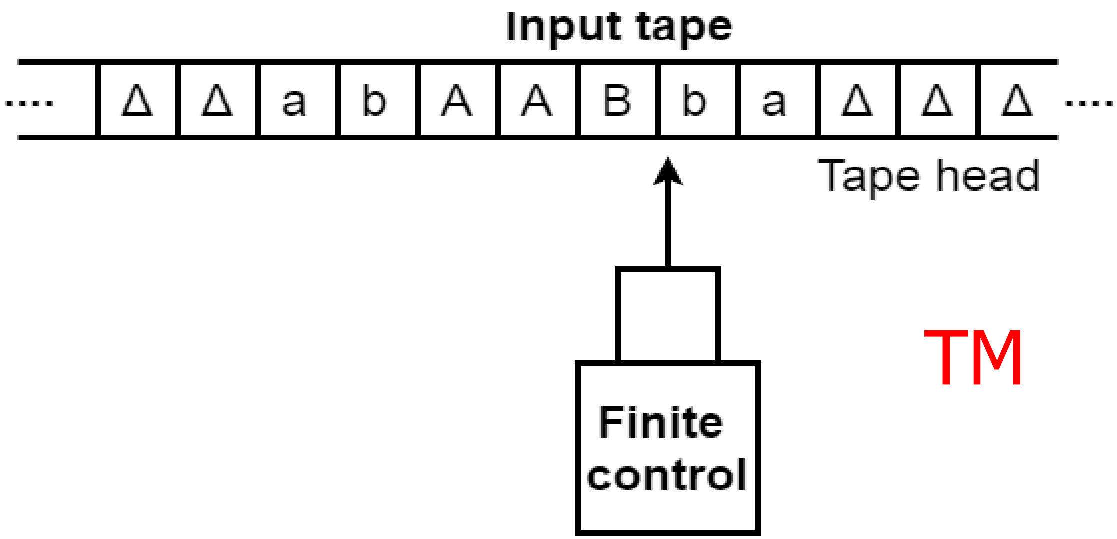
Finite automata	Devices with a finite amount of memory. Used to model "small" computers like lexical analyser.
Push-down automata	Devices with infinite memory that can be accessed in a restricted way(one direction) Used to model parsers, etc.
Turing Machines	Devices with infinite memory(both direction) Used to model any computer.
Time-bounded Turing Machines	Infinite memory, but bounded running time. Used to model any computer program that runs in a "reasonable" amount of time.

D

Architectural Representation of Automata



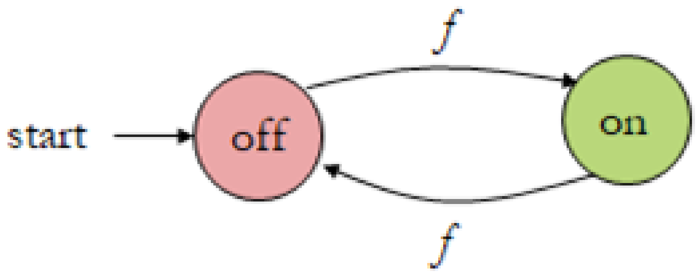




Dr.PS

Finite Automata

- What is this?



- There are states off and on, the automaton starts in off and tries to reach the "good state" on
- What sequences of fs lead to the good state?

  Answer: {f, fff, fffff, …} = {f^n: n is odd}
- This is an example of a deterministic finite automaton over alphabet {f}

Dr.PS

Deterministic Finite Automata

- A deterministic finite automaton (DFA) is a 5-tuple notation:

  A= $(\mathcal{Q}, \Sigma, \delta, q_0, F)$ where
  - $\mathcal{Q}$ is a finite set of states
  - Σ is an alphabet
  - $\delta: \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is a transition function
  - $q_0 \in \mathcal{Q}$ is the initial state
  - $F \subseteq \mathcal{Q}$ is a set of accepting states (or final states).
- In transition diagrams, the accepting states will be denoted by double loops

How a DFA processes strings

- Given an input string $x = a_1 a_2 \ldots a_n$, if

$\delta(q_0, a_1) = q_1,$

$\delta(q_1, a_2) = q_2,$

$\ldots,$

$\delta(q_{i-1}, a_i) = q_i,$

$\ldots,$

$\delta(q_{n-1}, a_n) = q_n,$ and $q_n \in F,$

then x is "accepted"; otherwise, "rejected."

- Every transition is deterministic.

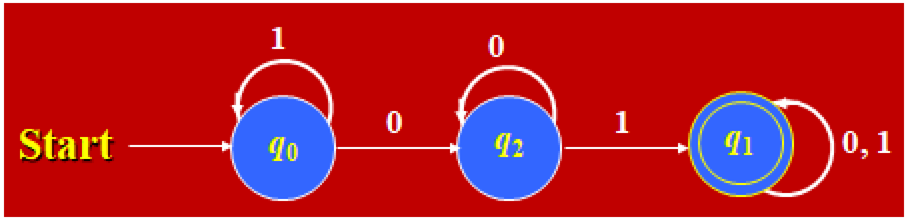
Designing DFA

- Design an FA *A* to accept the language

$$L = \{x01y \mid x \text{ and } y \text{ are any strings of 0's and 1's}\}.$$

- Examples of strings in *L*={<u>01</u>, 11<u>01</u>0, 100<u>01</u>1…}

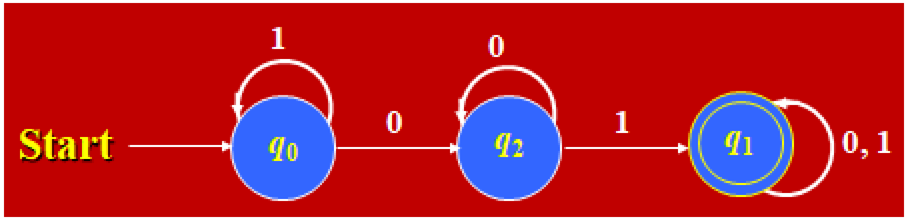
- Transition diagram



- Why is it deterministic?

Simpler Notations for DFA's

- Transition diagram
- `



- Transition Table

	0	1
$\rightarrow q_0$	q_2	q_0
$*q_1$	q_1	q_1
q_2	q_2	q_1

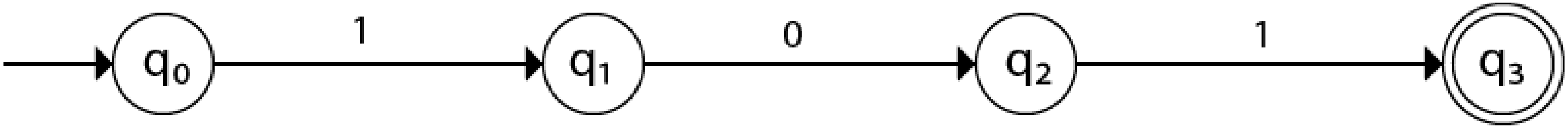
$\rightarrow$: initial state; $*$: final state

Dr.PS

Designing DFA Example-2

- Design a FA with ∑ = {0, 1} accepts the only input 101

Solution:



Dr.PS

Summary

- Types of Automata
- Pictorial representation of Automaton
- Introduction to Finite Automata
- Deterministic Finite Automata(DFA)
- Definition of DFA
- Notations of DFA
- Extended transition function of DFA
- Designing DFA-Examples

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

Non-deterministic Finite Automata(NFA)

THANK YOU.