Automata and Language Theory

Chapter 6 (Finite Automata)

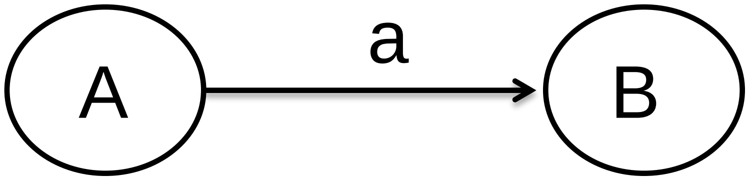
Part 1

Dr. Doaa Shebl
Faculty of Computers and Artificial Intelligence
Beni-Suef University

Chapter 6- Finite Automata

Deterministic Finite Automaton (DFA)

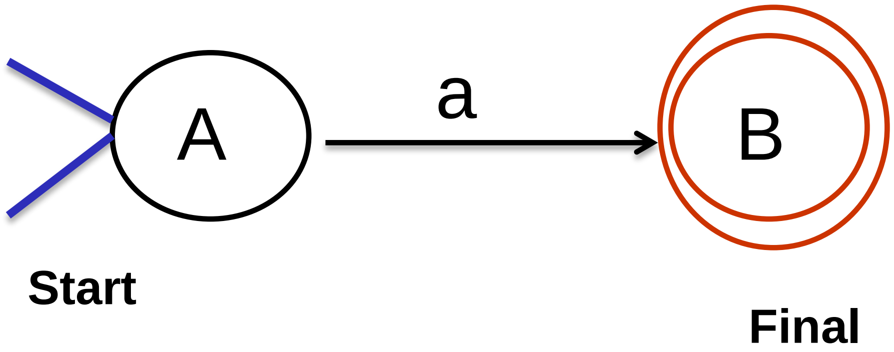
- An informal definition
  - A diagram with a finite number of states represented by circles
  - An arrow points to one of the states, the unique *start state*
  - Double circles mark any number of the states as *accepting states*



$A \rightarrow a\ B$

Chapter 6- Finite Automata

Deterministic Finite Automaton (DFA)



$A \rightarrow a\ B$

Chapter 6- Finite Automata

- For every state, for every symbol in Σ, there is exactly one arrow labeled with that symbol going to another state (or back to the same state).

- Given any string over Σ, a DFA can read the string and follow its state-to-state transitions.

- At the end of the string, if it is in an accepting state, we say it accepts the string. Otherwise it rejects.

- The language defined by a DFA is the set of strings in Σ^* that it accepts.

Chapter 6- Finite Automata

Definition

The state diagram of a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is a labeled graph G defined by the following conditions:

i) The nodes of G are the elements of Q.

ii) The labels on the arcs of G are elements of Σ.

iii) q_0 is the start node, depicted $\rightarrow\bigcirc$.

iv) F is the set of accepting nodes; each accepting node is depicted $\circledcirc$.

v) There is an arc from node q_i to q_j labeled a if $\delta(q_i, a) = q_j$.

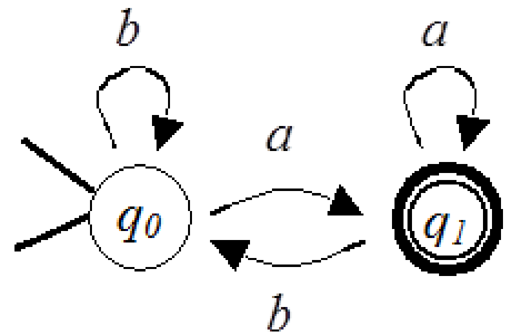
vi) For every node q_i and symbol $a \in \Sigma$, there is exactly one arc labeled a leaving q_i.

δ is the transition function

A function $\delta(q,a)$ that takes the current state q and next input symbol a, and returns the next state

Chapter 6- Finite Automata

Example 1:



- This DFA defines $\{xa \mid x \in \{a,b\}^*\}$
- Formally, $M = (Q, \Sigma, \delta, q_0, F)$, where
  - $Q = \{q_0, q_1\}$
  - $\Sigma = \{a,b\}$
  - $F = \{q_1\}$
  - *Start sate* q_0

Give the transition table

	a	**b**
q_0	q_1	q_0
q_1	q_1	q_0

  - $\delta(q_0,a) = q_1$, $\delta(q_0,b) = q_0$, $\delta(q_1,a) = q_1$, $\delta(q_1,b) = q_0$

- We could just say $M = (\{q_0,q_1\}, \{a,b\}, \delta, q_0, \{q_1\})$

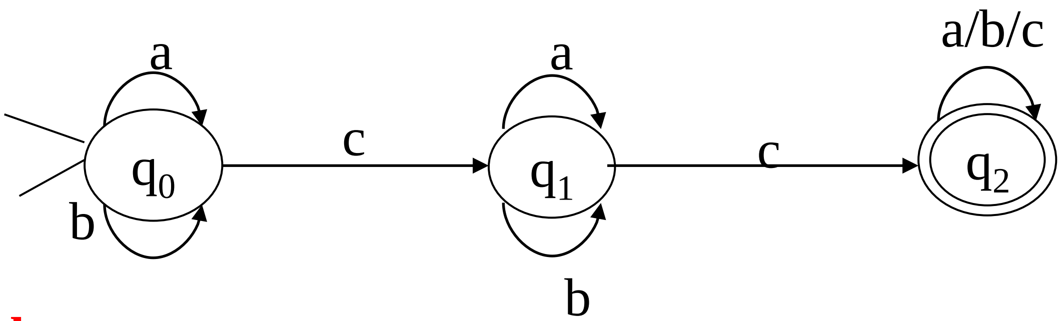
Example 2:

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

Start state is q_0

$F = \{q_2\}$

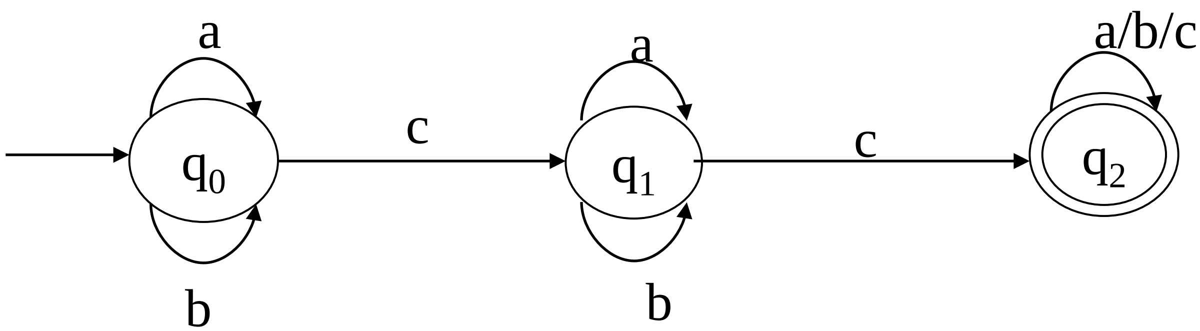


	a	b	c
q_0	q_0	q_0	q_1
q_1	q_1	q_1	q_2
q_2	q_2	q_2	q_2

- Since δ is a function, at each step M has exactly one option.

Chapter 6- Finite Automata

Example 3:



Which of the string acccb and aac accepted by Machine M ???

Acccb

$\vdash [q_0, \text{acccb}]$

$\vdash [q_0, \text{cccb}]$

$\vdash [q_1, \text{ccb}]$

$\vdash [q_2, \text{cb}]$

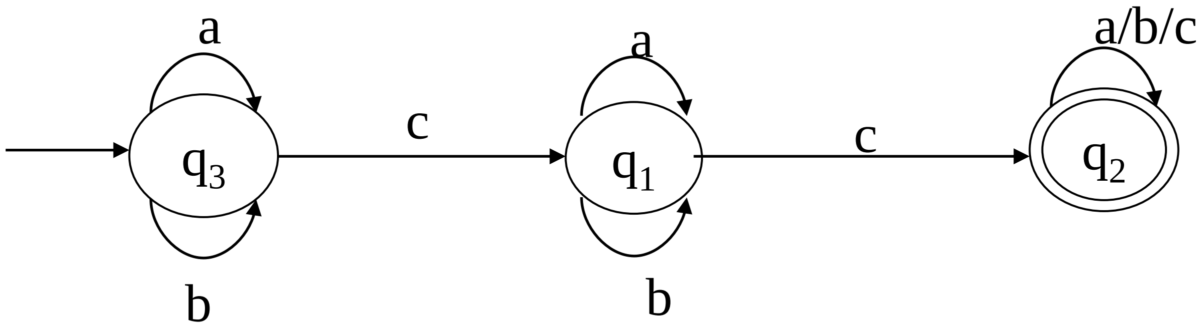
$\vdash [q_2, \text{b}]$

$\vdash [q_2, \lambda]$

accepted

Chapter 6- Finite Automata

Example 3:



aac

$\vdash$ $[q_0, aac]$

$\vdash$ $[q_0, ac]$

$\vdash$ $[q_0, c]$

$\vdash$ $[q_1, \lambda]$

rejected

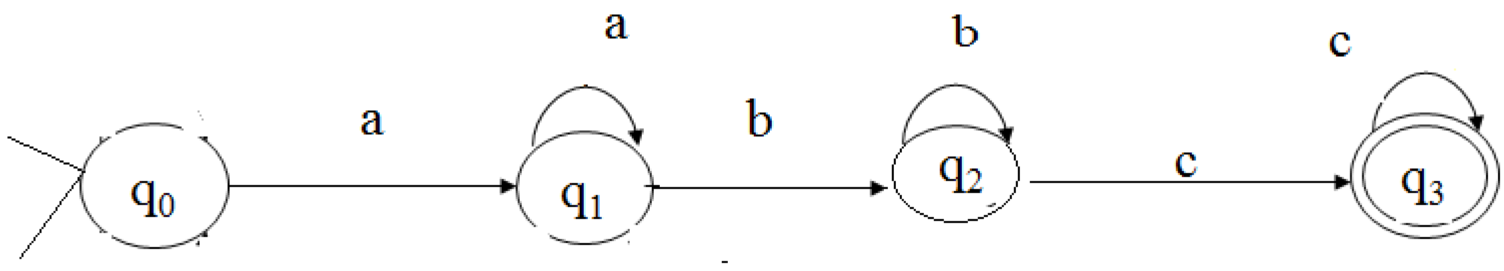
Accepts those strings that contain at least two c's

Chapter 6- Finite Automata

Example 4:

Give finite automata for the set of strings over {a, b, c } which all the a's precede the b's, which in turn precede the c's. without the null string .

$a^+ b^+ c^+$



$$G: q_0 \rightarrow aq_1$$

$$q_1 \rightarrow a\, q_1 \,/\, b\, q_2$$

$$q_2 \rightarrow b\, q_2 \,/\, c\, q_3$$

$$q_3 \rightarrow c\, q_3 \,/\, \lambda$$

Chapter 6- Finite Automata

Example 5:

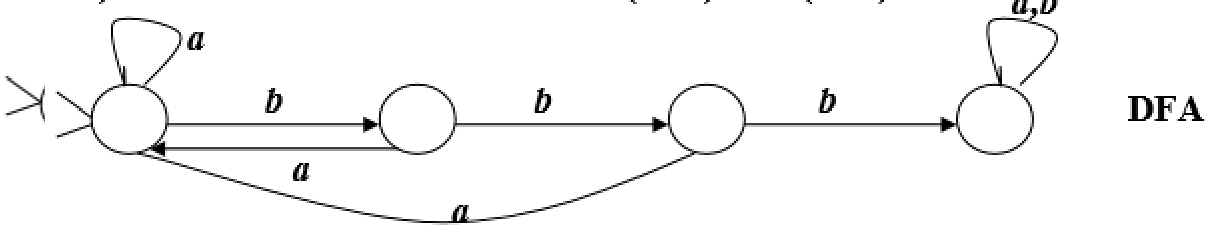
Give a finite automata for the set of strings over {a , b}:

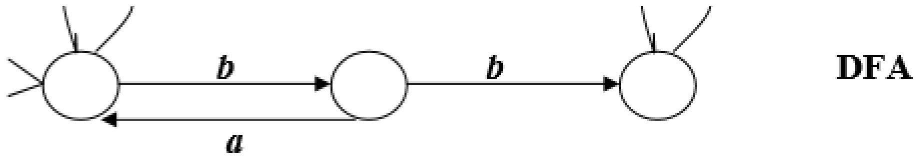
i) that contain bb. ii) that contain bbb.

Solution

i) $(a \cup b)^* \, bb \, (a \cup b)^*$

ii) **that contain bbb.** $(a \cup b)^* \, bbb \, (a \cup b)^*$

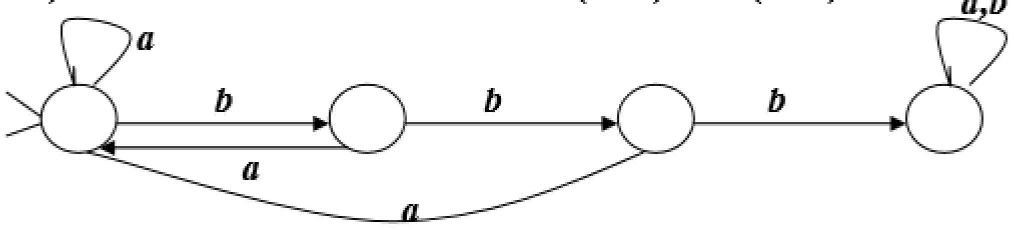




Chapter 6- Finite Automata

Example 5:

ii) **that contain bbb.** $(a \cup b)^* \, bbb \, (a \cup b)^*$

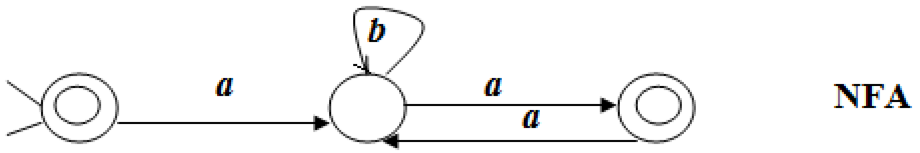


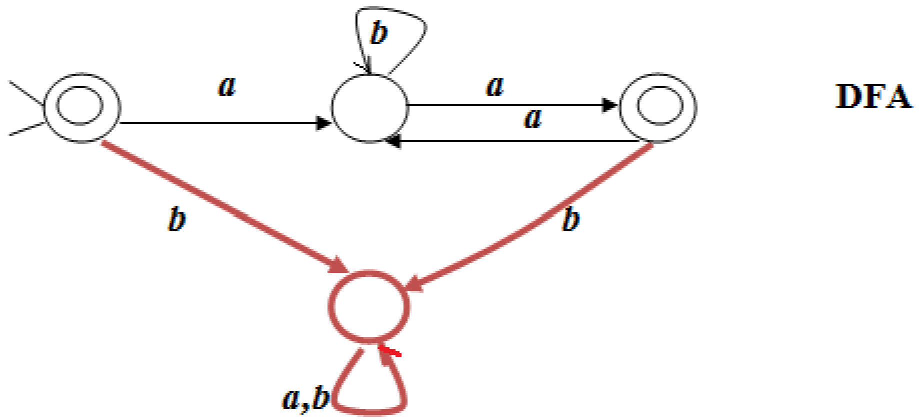
DFA

Chapter 6- Finite Automata

Example 6:

Give a state diagram of NFA and DFA for the following: (ab^*a)*.





DFA Applications

We have seen how DFAs can be used to define formal languages. In addition to this formal use, DFAs have practical applications. DFA-based pieces of code lie at the heart of many commonly used computer programs.

Chapter 6- Finite Automata

DFA Applications

- Programming language processing
  - Scanning phase: dividing source file into "tokens" (keywords, identifiers, constants, etc.), skipping whitespace and comments
- Command language processing
  - Typed command languages often require the same kind of treatment
- Text pattern matching
  - Unix tools like awk, egrep, and sed, mail systems like ProcMail, database systems like MySQL, and many others

Chapter 6- Finite Automata

More DFA Applications

- Signal processing
  - Speech processing and other signal processing systems use finite state models to transform the incoming signal
- Controllers for finite-state systems
  - Hardware and software
  - A wide range of applications, from industrial processes to video games