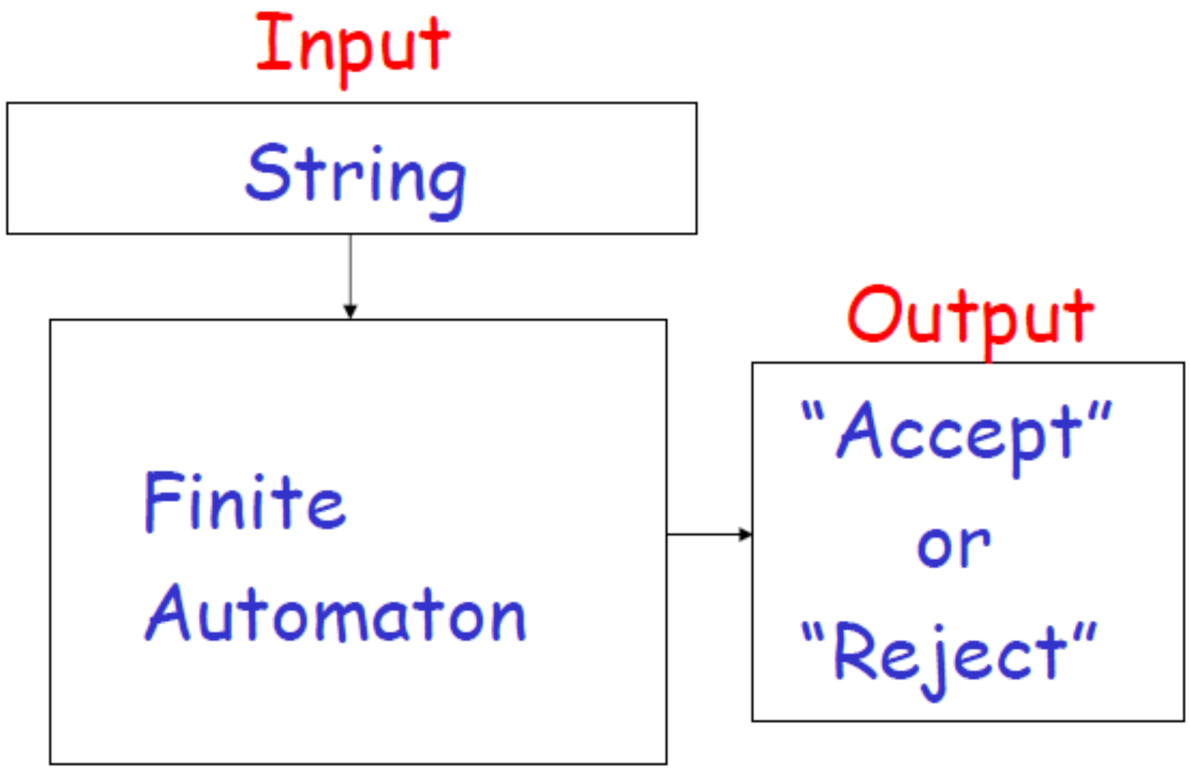
CS311 Formal Language and Automata

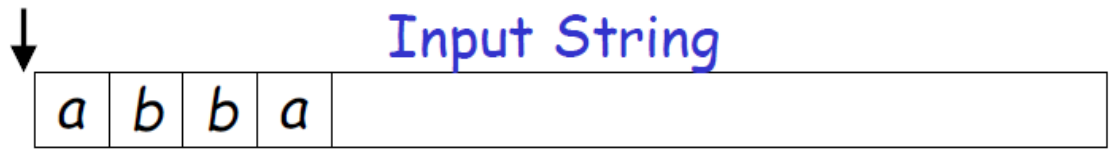
Tingting Chen
Computer Science
Cal Poly Pomona

Finite Automata
Deterministic Finite Accepter (DFA)



How the Automata work

- At the beginning, automaton is assumed to be in an initial state q_0, with the input control about to read the leftmost symbol of the input string (has not read it yet).



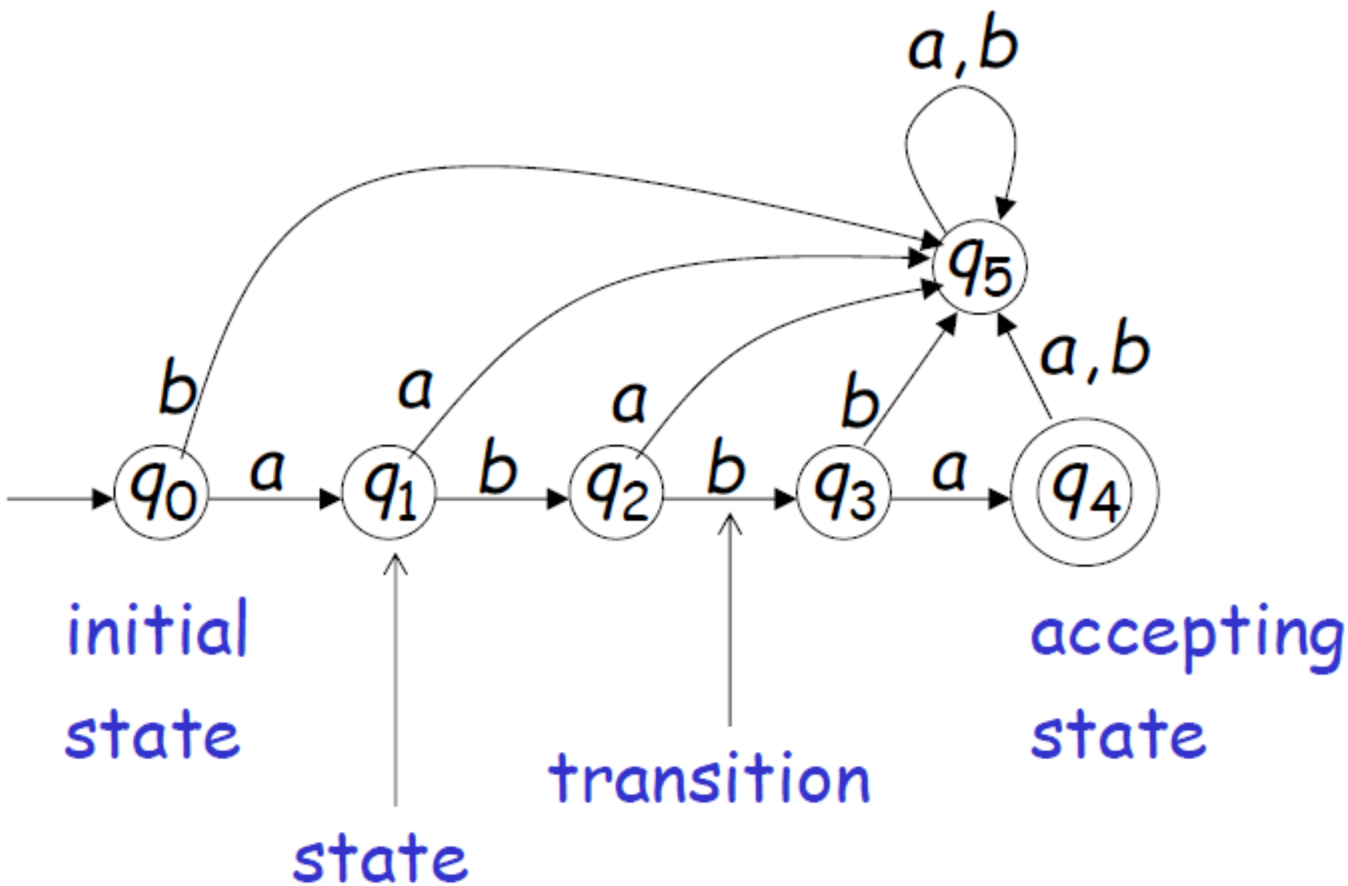
- Then in each move, automaton advances the input control one place at a time to the right, on the input string. The state of the automata will be updated (changed) to the next state. The transition is determined by transition function.

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

- When the end of string is reached, the string is accepted if at that time the automata is in one of the final (acceptance) states.

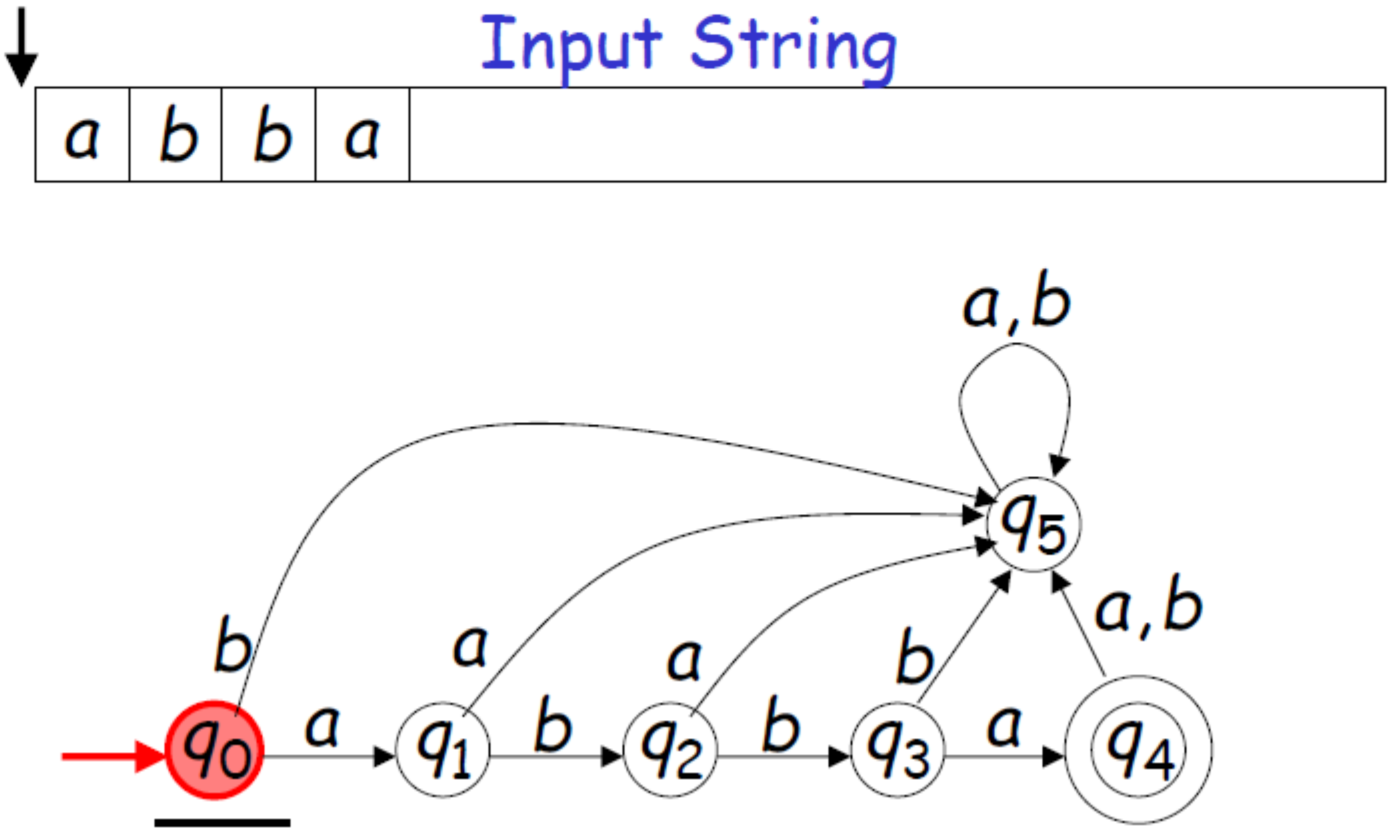
Transition Graph

- Vertices for states, and edges for transitions.
- Final state: two circles.
- Based on transition function.



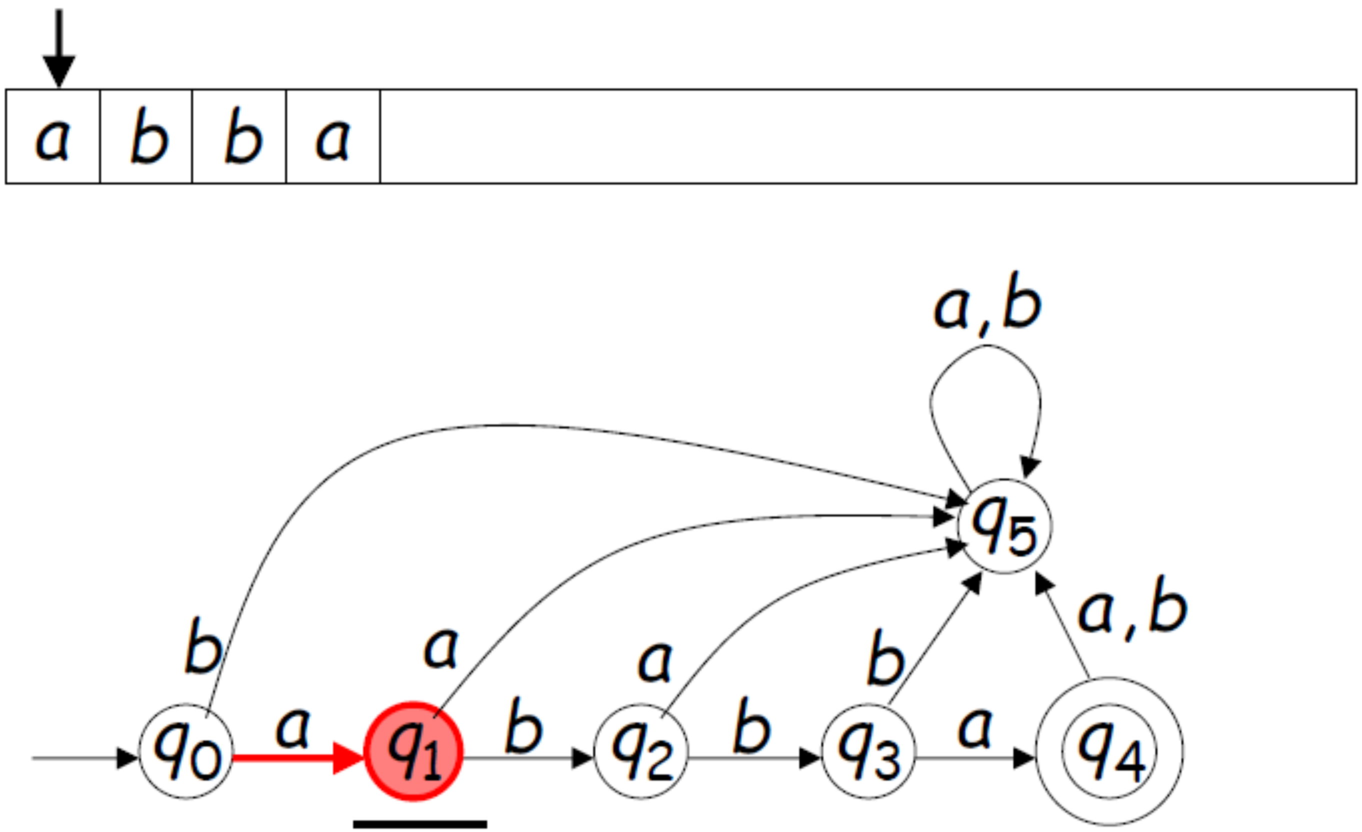
Example – Acceptance Case

- Initial configuration



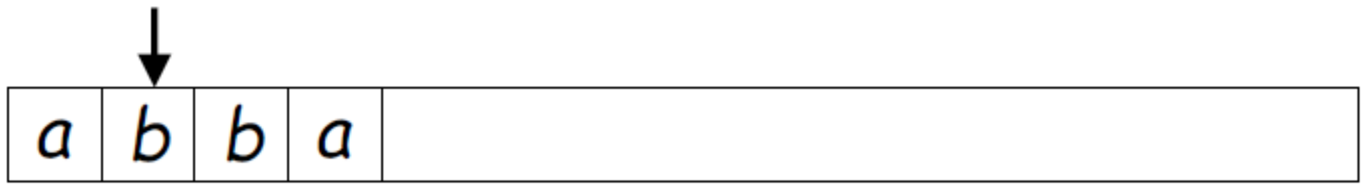
Example – Acceptance Case

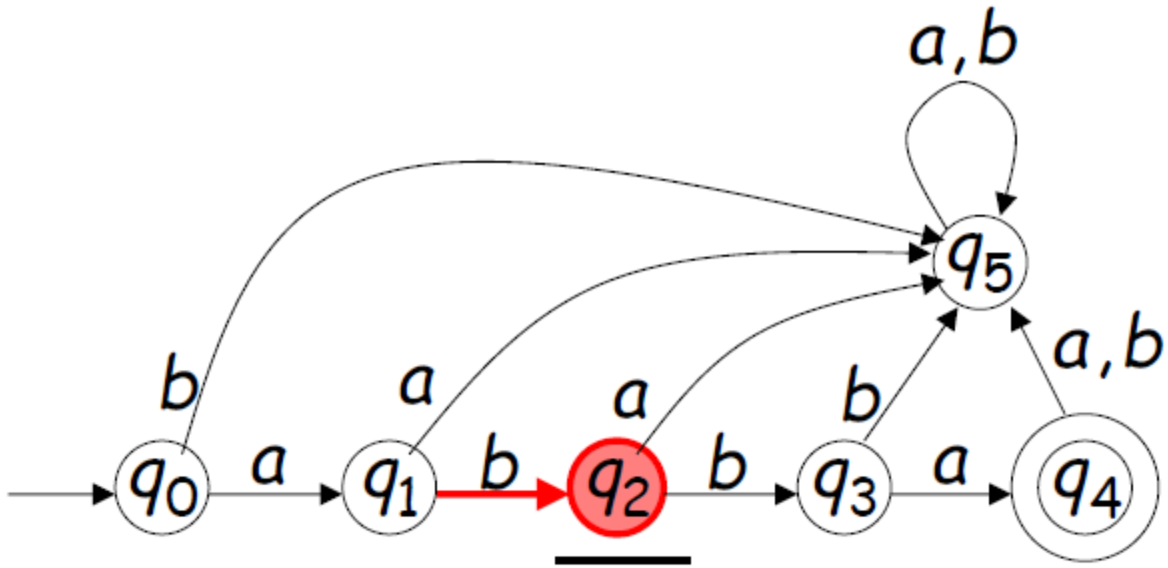
- Reading the input



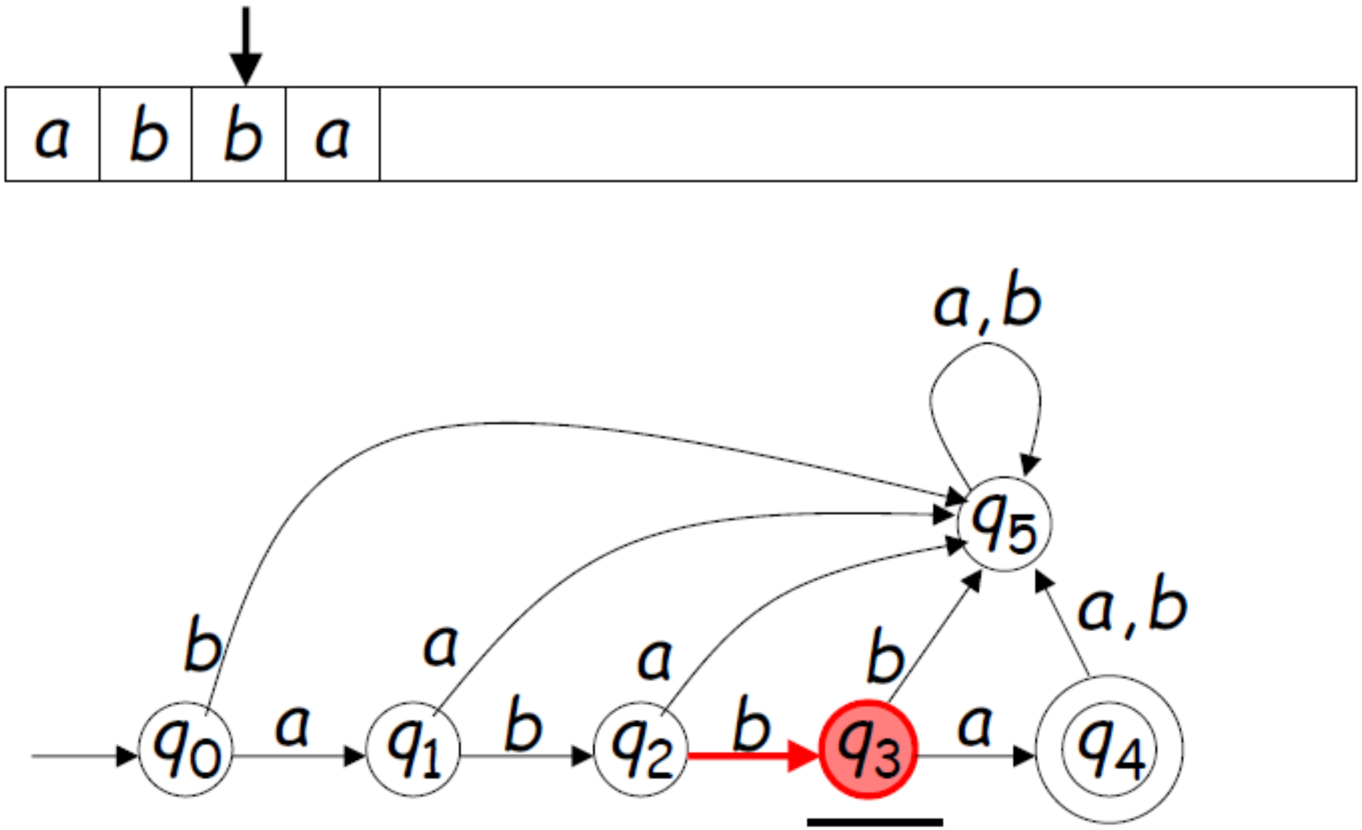
Example – Acceptance Case

- Reading the input

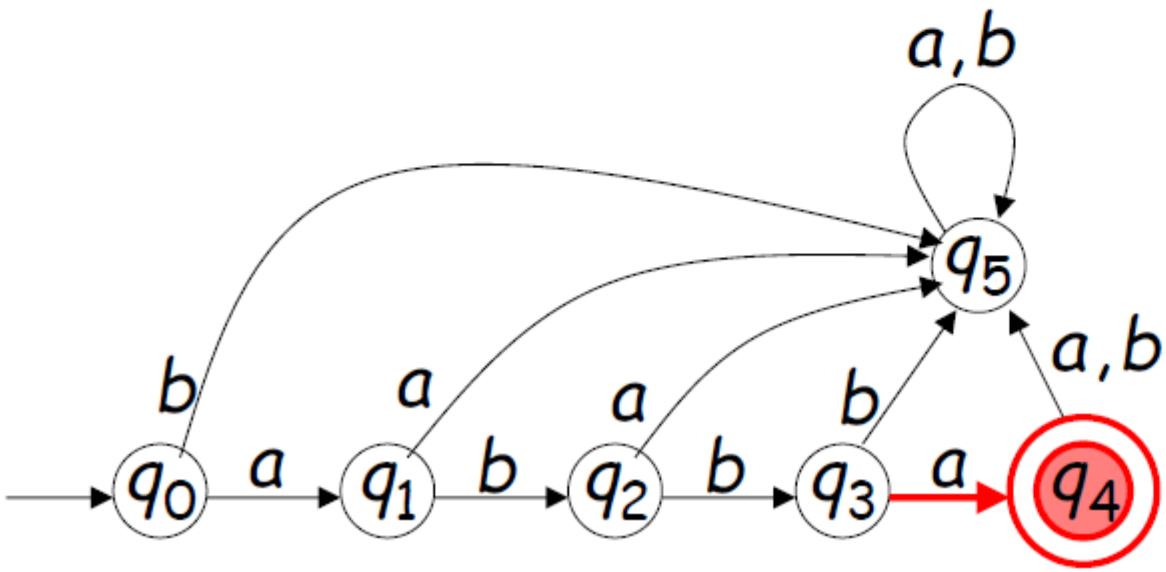




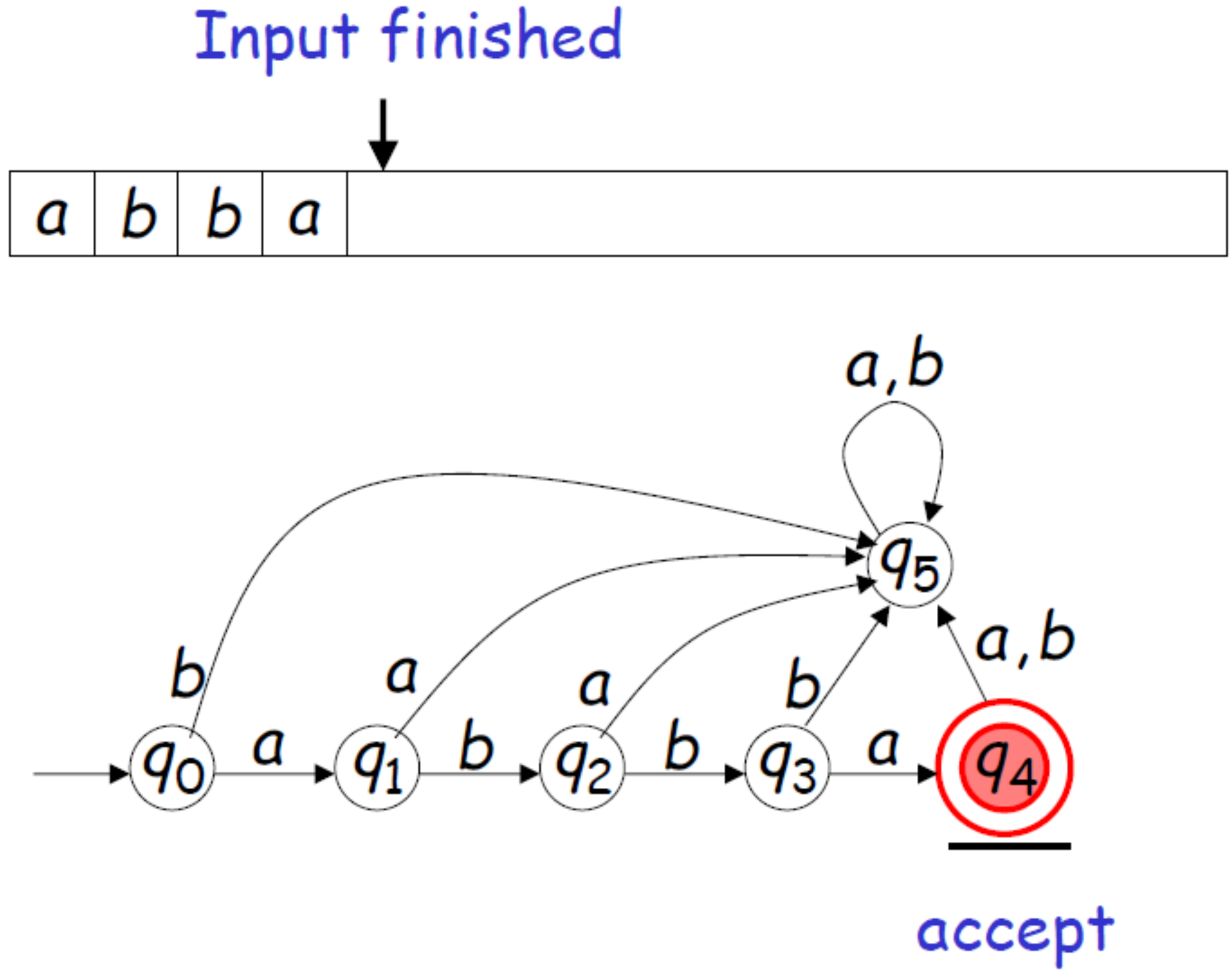
Example – Acceptance Case



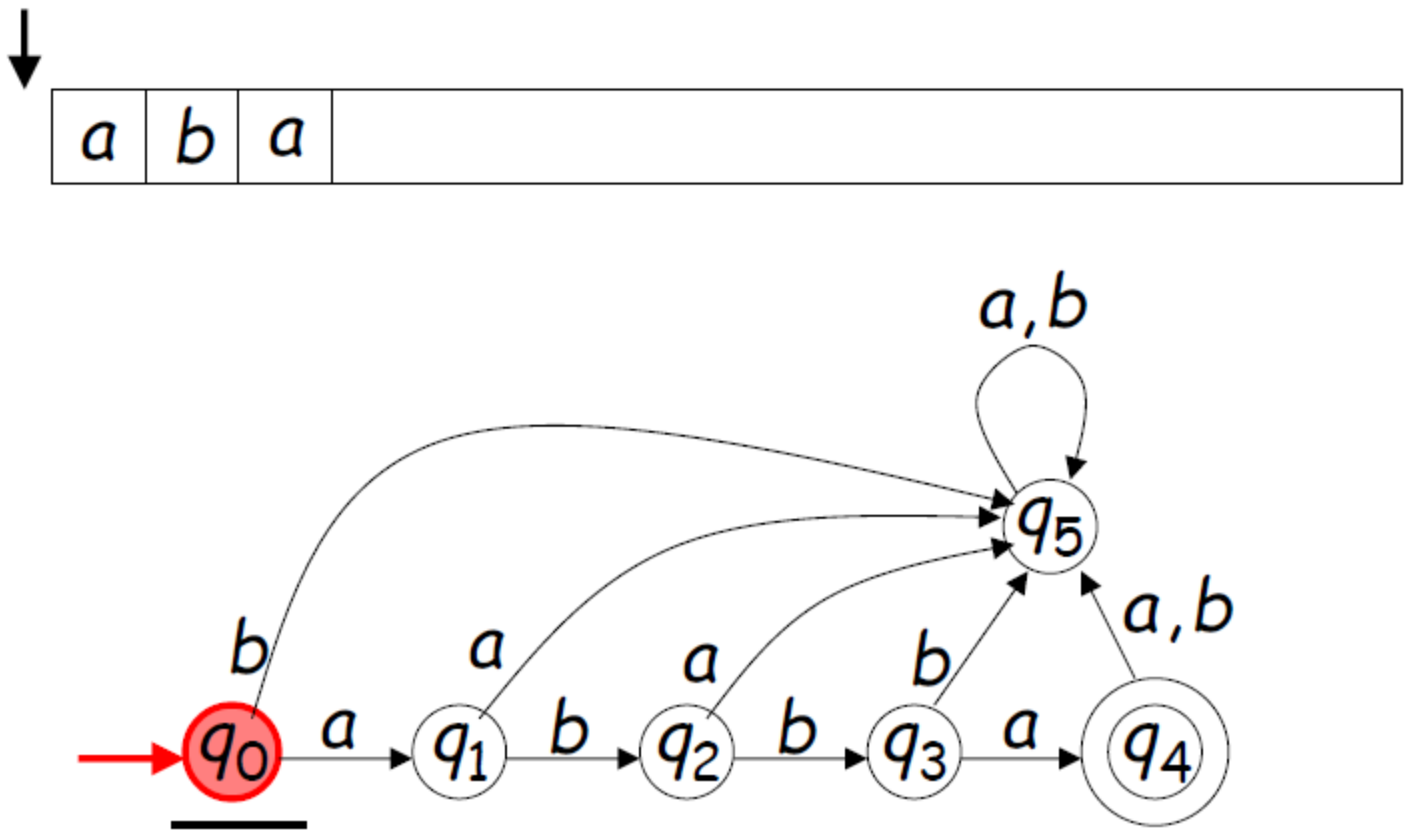
Example – Acceptance Case



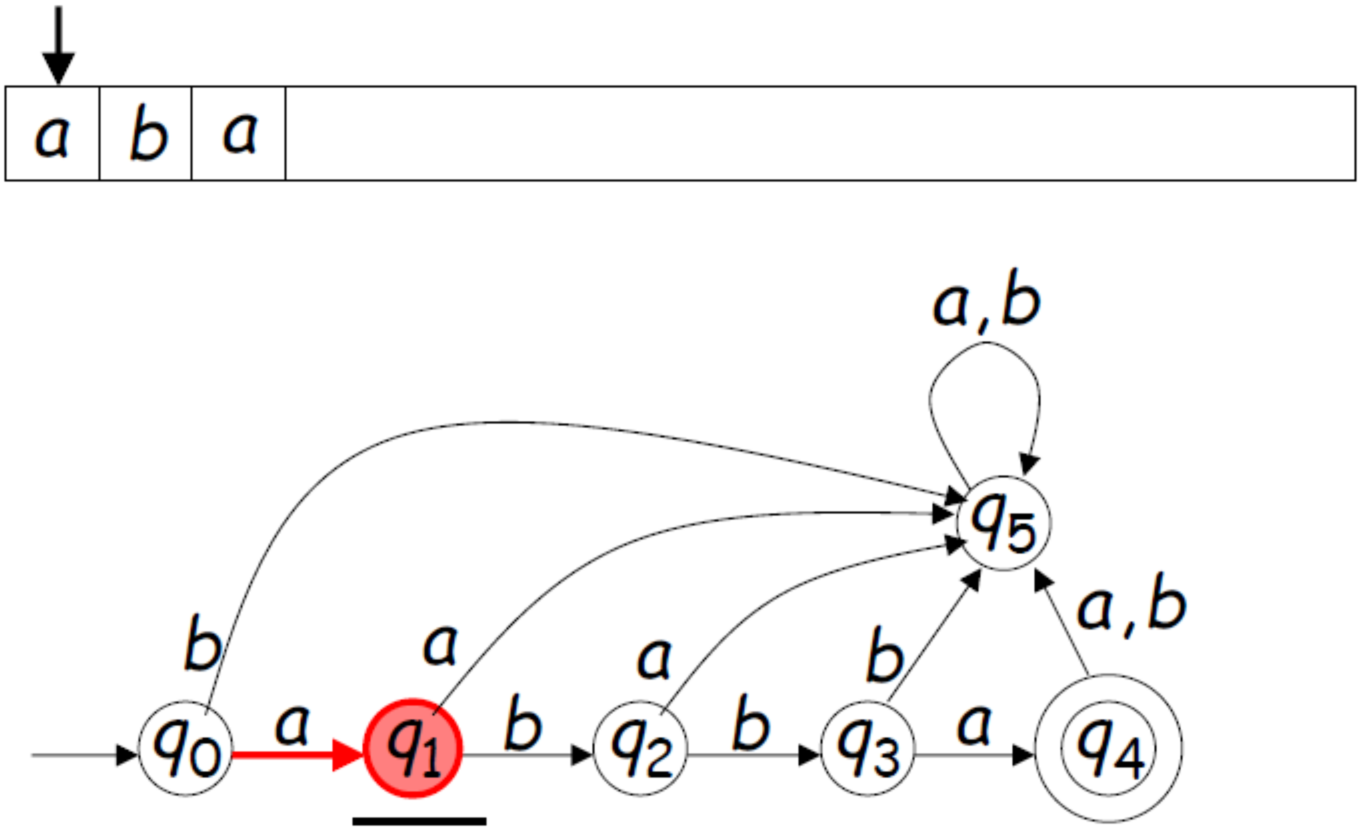
Example – Acceptance Case



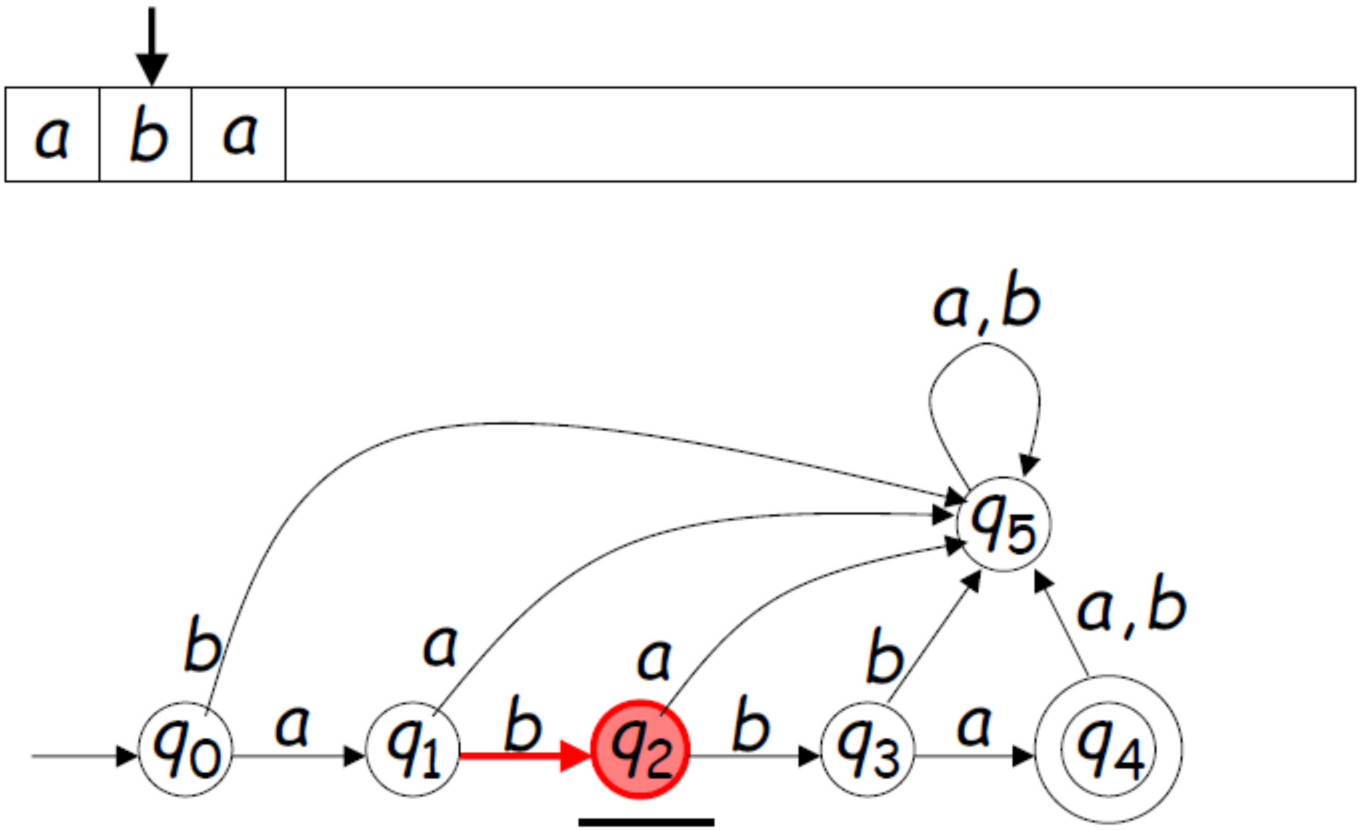
Rejection



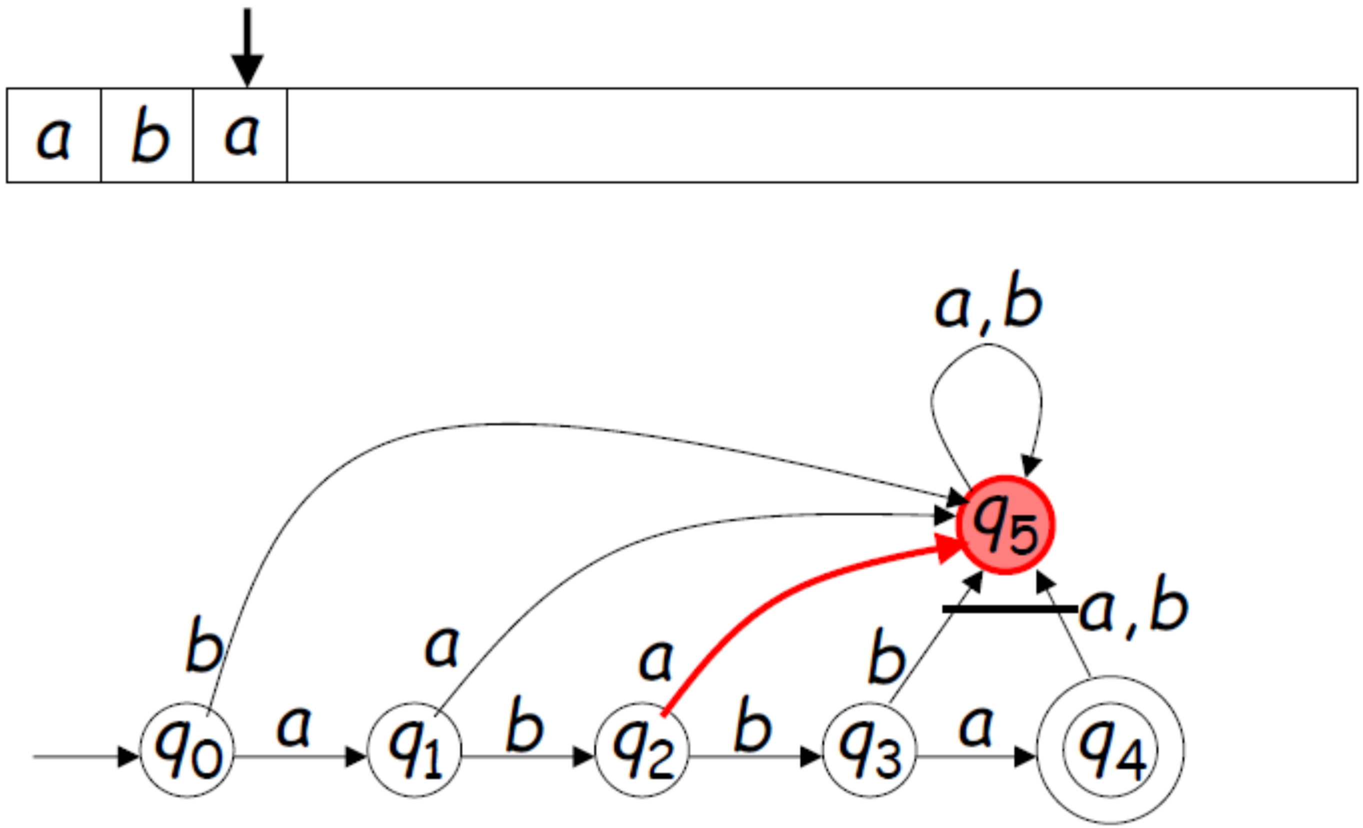
Rejection



Rejection

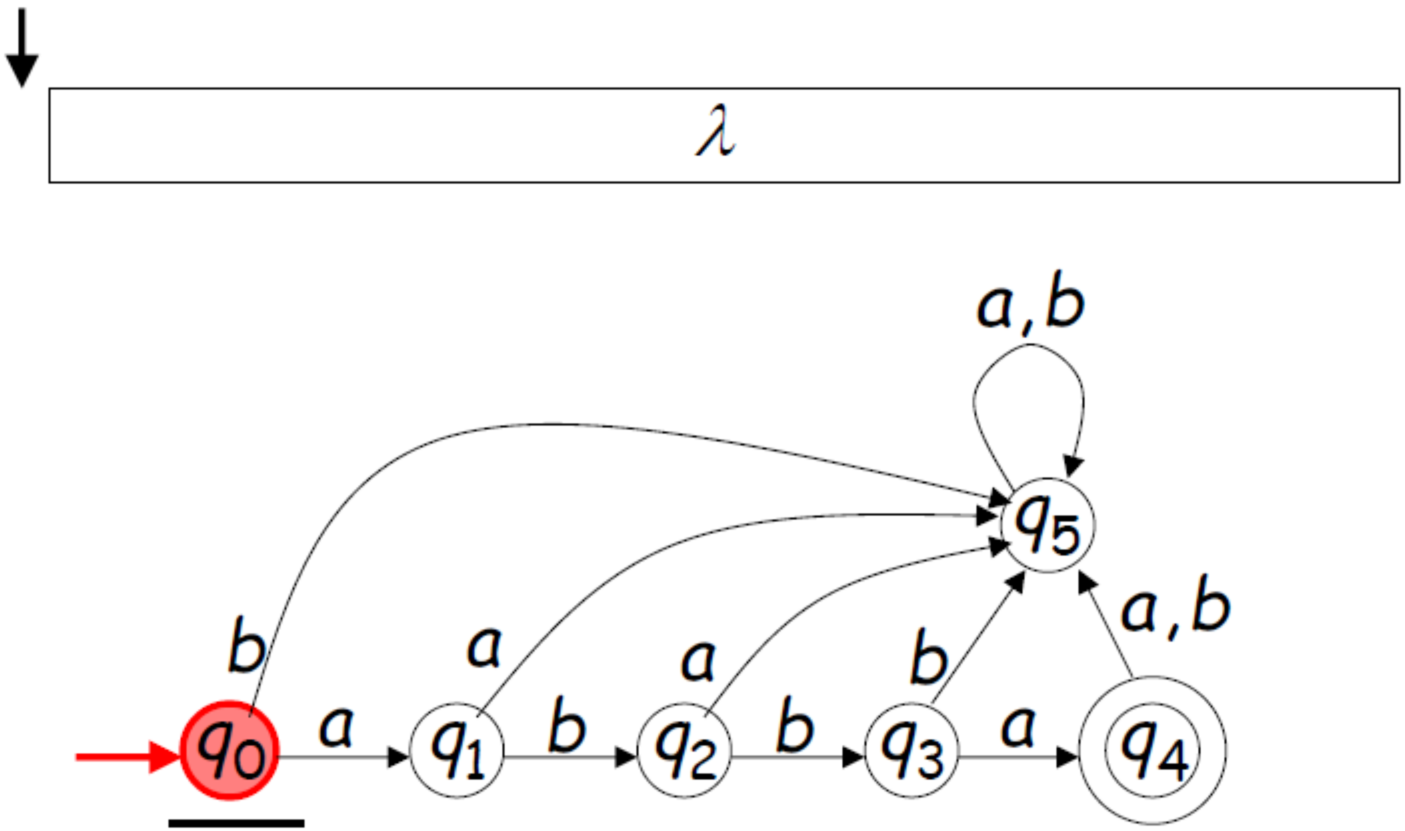


Rejection



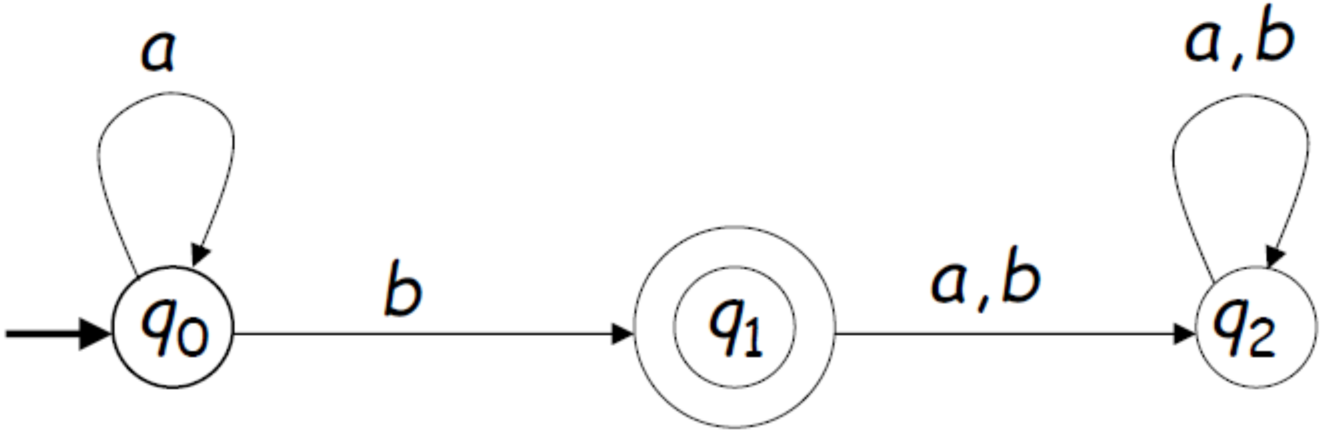
Acceptance or Rejection?

- Input: empty string



Language Recognition

- Given a particular DFA with its transition graph, you can tell what language it accepts.
- Example:



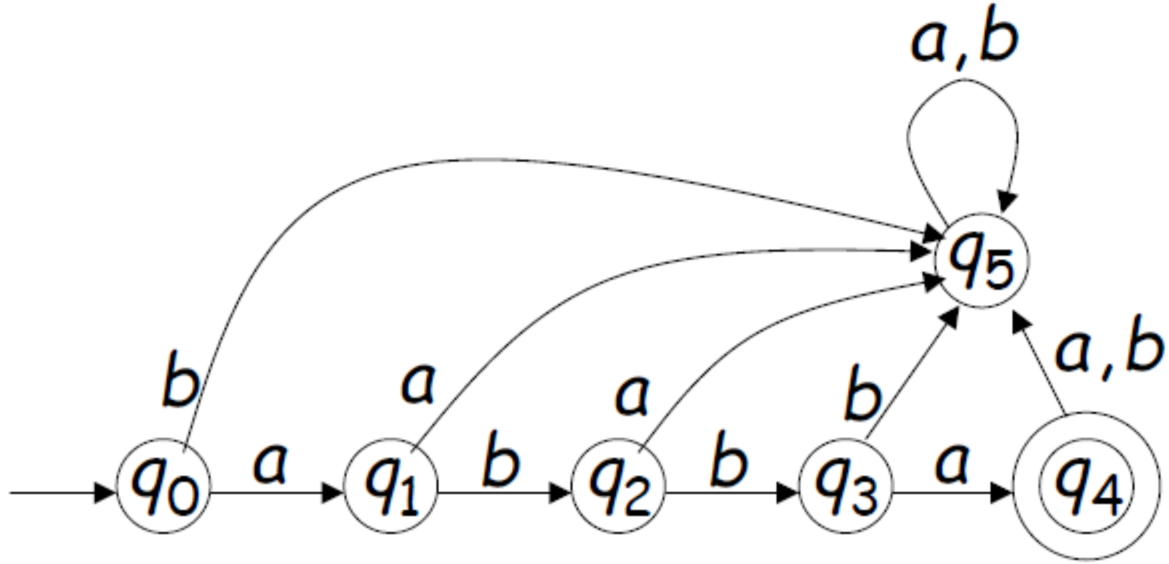
$\{a^n b, n >= 0\}$

Formally, What defines a DFA?

- Defined by a quintuple: $M = (Q, \Sigma, \delta, q_0, F)$
  - Q is a finite, nonempty set of states
  - Σ is finite set of input symbols called alphabet
  - $\delta: Q \times \Sigma \to Q$ is the transition function

    transition function can be regarded as “program”
  - $q_0 \in Q$ is the initial state
  - $F \subseteq Q$ is a set of final or “accepting” states

DFA -- Example



- $Q = \{q_0, q_1, q_2, q_3, q_{4,} q_5\}$
- $\Sigma = \{a, b\}$
- q_0
- $F = \{q_4\}$
- $\delta(q_0, a) = q1$, $\delta(q_0, b) = q_5$, $\delta(q_1, a) = q_5$, $\delta(q_1, b) = q_2$, $\delta(q_2, a) = q5$, $\delta(q_2, b) = q_3$, $\delta(q_3, a) = q_4$, $\delta(q_3, b) = q_5$, $\delta(q_4, a) = q5$, $\delta(q_4, b) = q_5$, $\delta(q_5, a) = q_5$, $\delta(q_5, b) = q_5$

Deterministic Finite Accepter -- Example

Example:

$M = (\{q_0, q_1, q_2\}, \{a,b\}, \delta, q_0, \{q_1\})$

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

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

δ is the transition function (see next slide)

q_0 is the initial state

$\{q_1\}$ is the set of final states

Transition table

The transition function of a finite automaton can be represented by a table:

state	input	next state
q_0	a	q_0
q_0	b	q_1
q_1	a	q_2
q_1	b	q_2
q_2	a	q_2
q_2	b	q_2

$M = (\{q_0, q_1, q_2\}, \{a,b\}, \delta, q_0, \{q_1\})$

The transition graph of M?

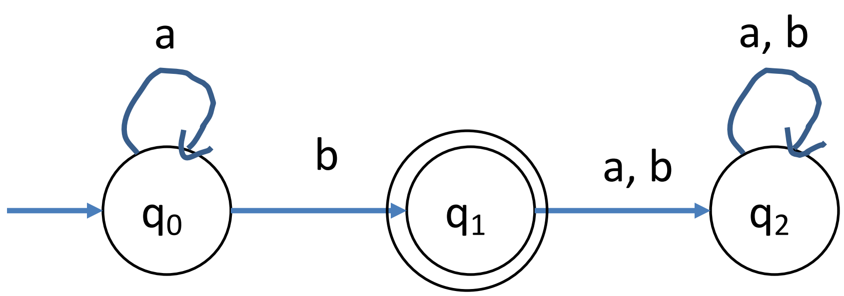
State transition table

Another format of a *state transition table*:

	a	b
$\rightarrow$ $\mathbf{q_0}$	q_0	q_1
$\mathbf{q_1}$	q_2	q_2
$\mathbf{q_2}$	q_2	q_2

From State transition table to Transition graph

	a	b
$\rightarrow \mathbf{q_0}$	q_0	q_1
$\mathbf{q_1}$	q_2	q_2
$\mathbf{q_2}$	q_2	q_2



Extended Transition Function

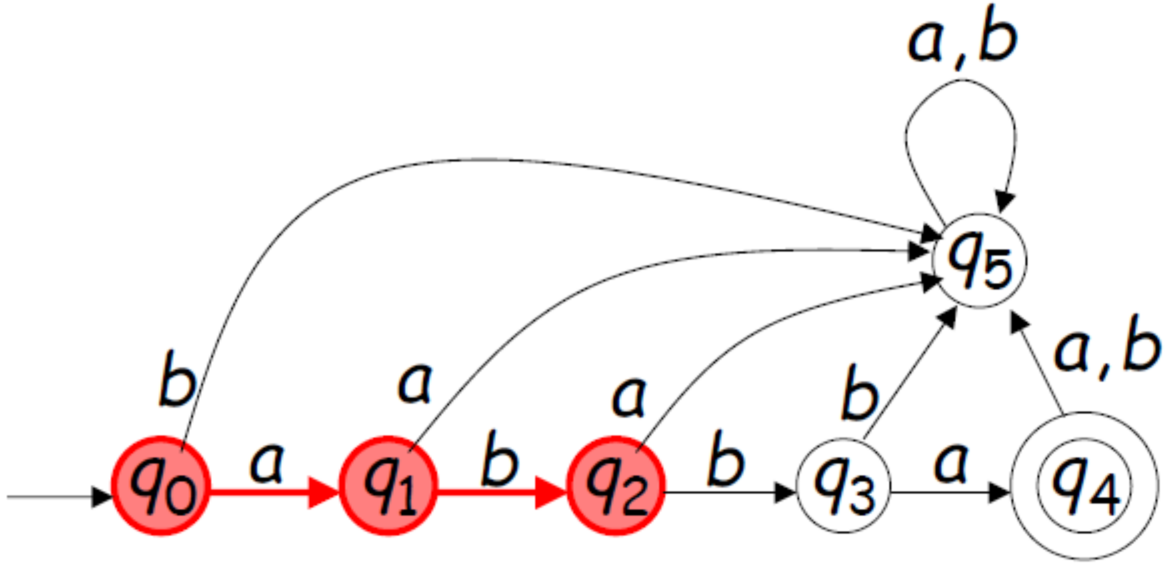
- The extended transition function is represented by:

  $$\delta^* : Q \times \Sigma^* \rightarrow Q$$

- It means the second input to the function is an element in Σ^*, which is a string!
- Q will represent the state the automaton will be in after reading the entire string instead of a single character

Extended Transition Function Example

$$\delta^*(q_0, ab) = q_2$$

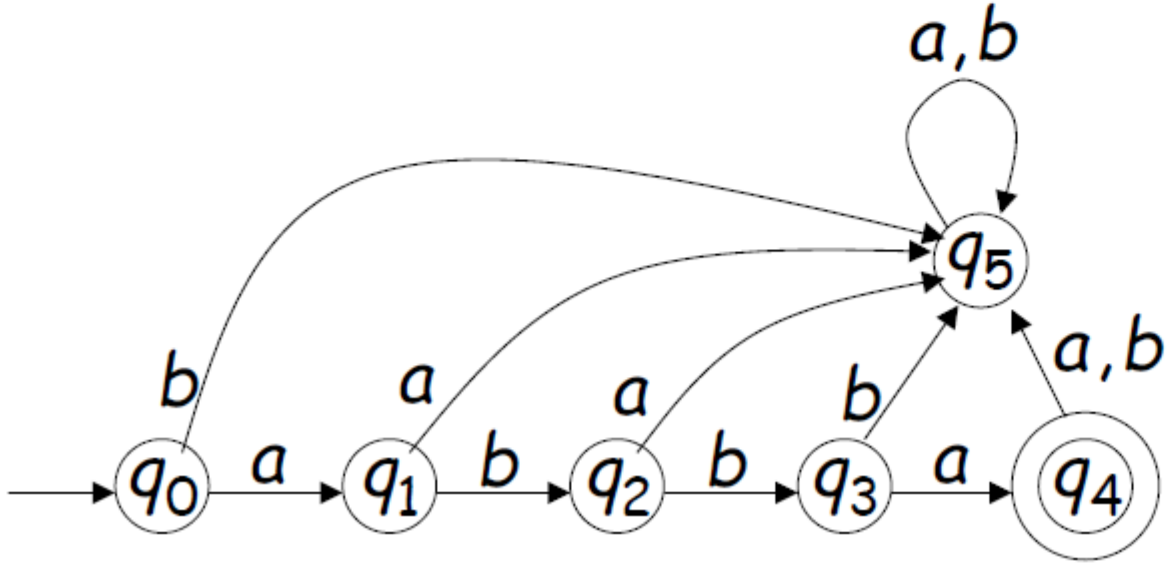


Extended Transition Function Example

$\delta^*(q_0, ab) = ?$

$\delta^*(q_2, babbabba) = ?$

$\delta^*(q_0, abbabbabba) = ?$



Extended Transition Function Example

$\delta^*(q_0,ab) =$

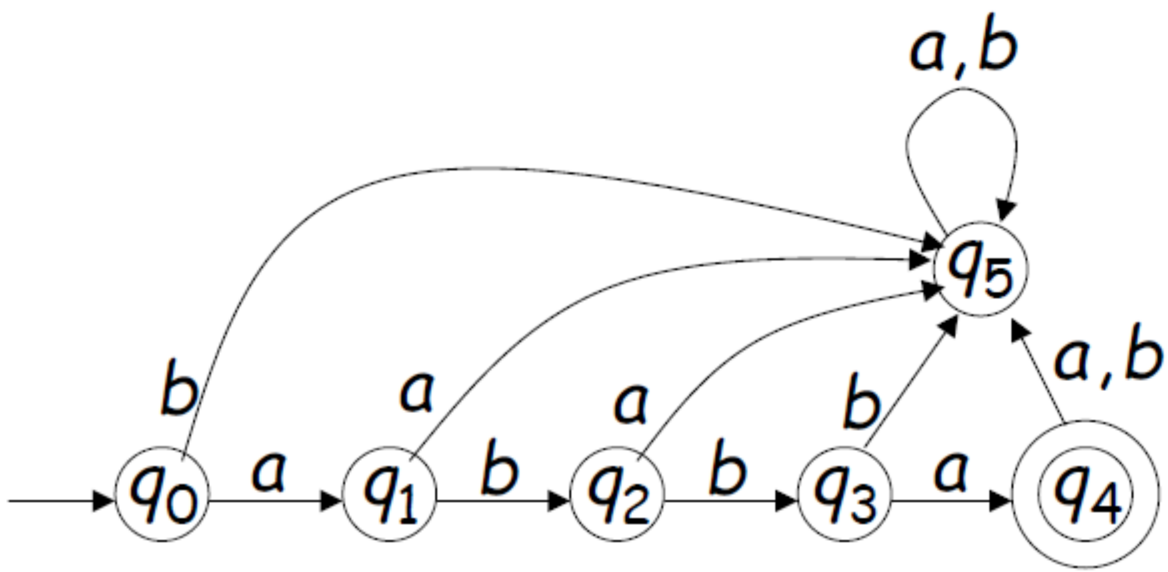
$\delta\,(\delta^*(q_0,a), b) =$

$\delta\,(\delta\,(\delta^*(q_0, \lambda),a), b) =$

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

$\delta\,(q_1, b)=$

q_2

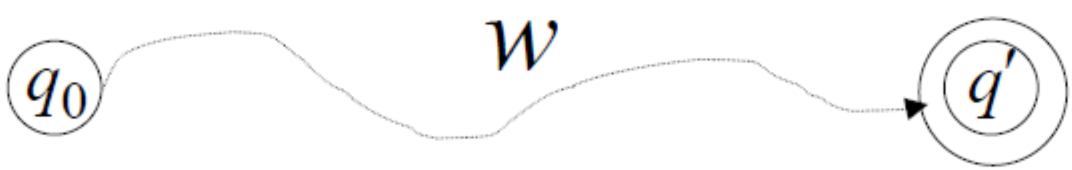


Language accepted by DFA, formally…

For a DFA $M = (Q, \Sigma, \delta, q_0, F)$,

The language accepted (or recognized) by M is the set of all strings on Σ that are accepted by M

$L(M) = \{w \in \Sigma^* : \delta^* (q_0, w) \in F\}$



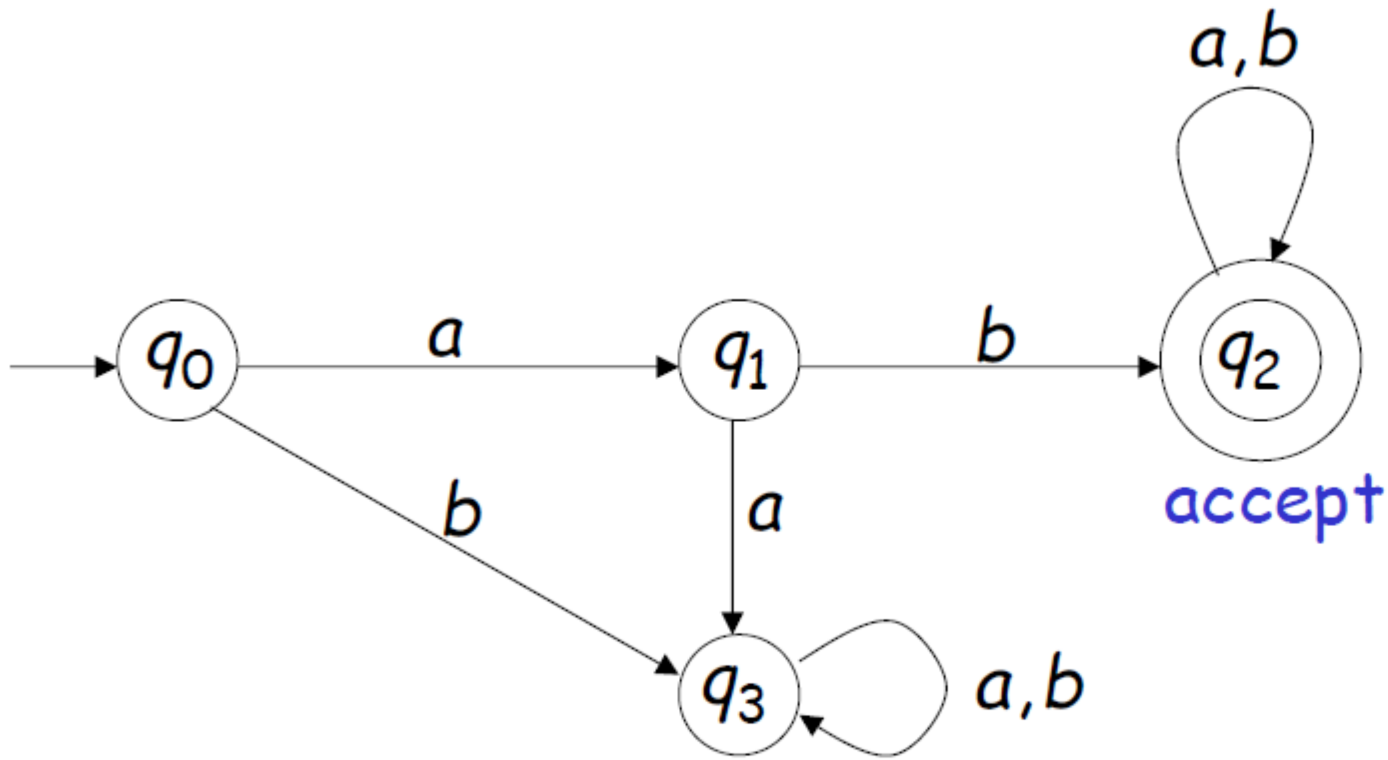
$q' \in F$

Language Rejected by DFA

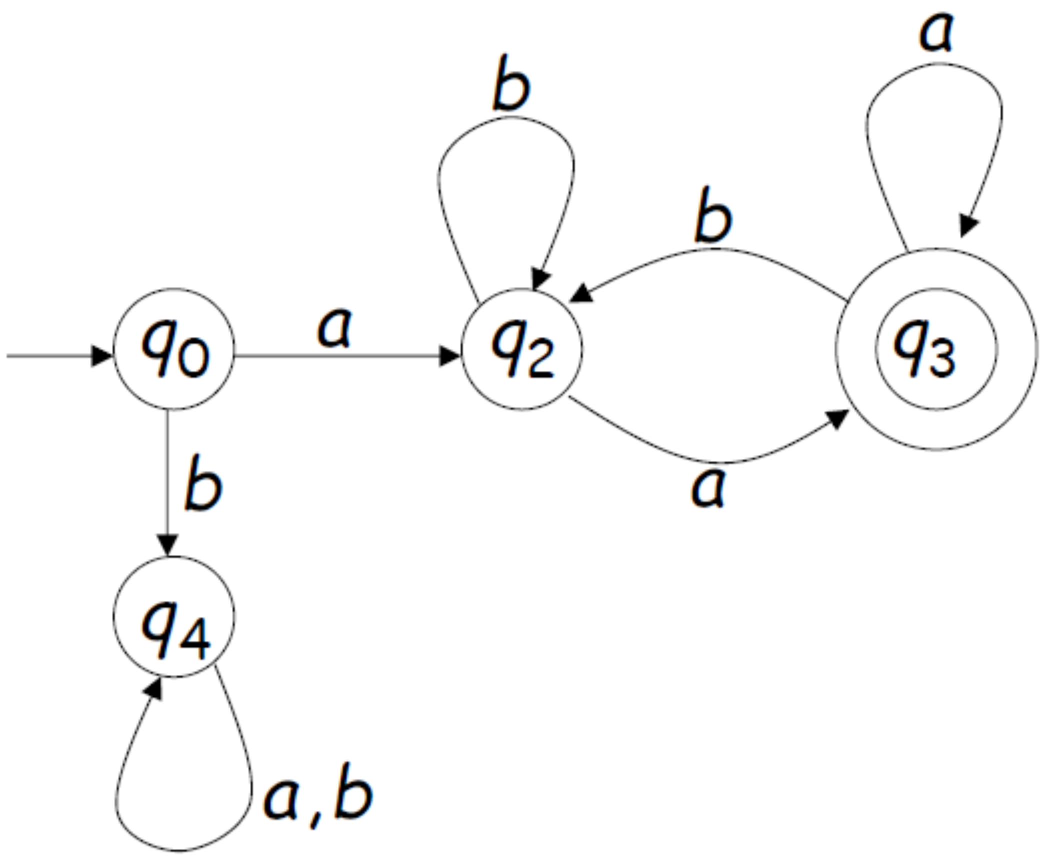
$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$

$q' \notin F$

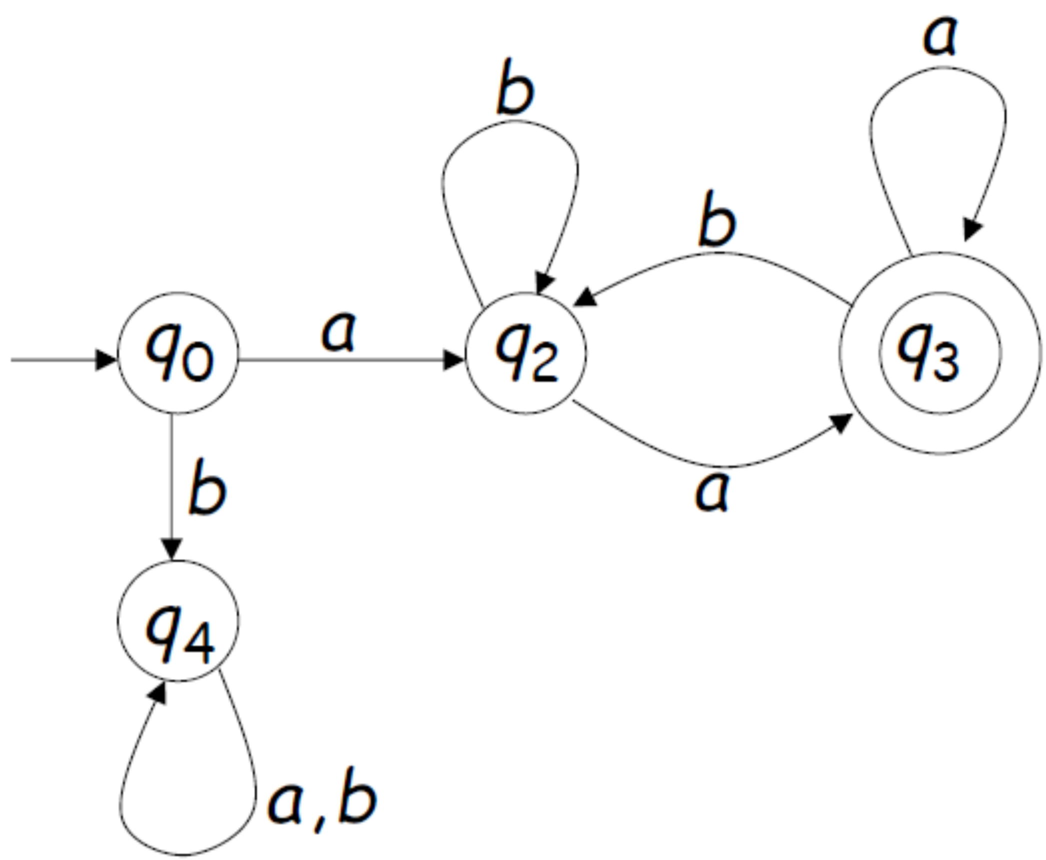
L(M) = ?



L(M) = ?



L(M) = ?



L(M) = {awa: w $\in \{a, b\}^*$}

Regular Languages

Prove that {all string s without substring *001*} is a regular language.