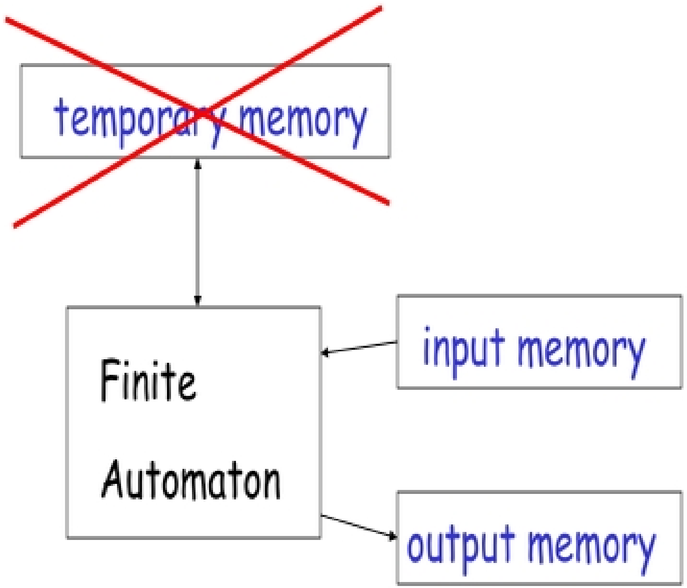
Finite Automata

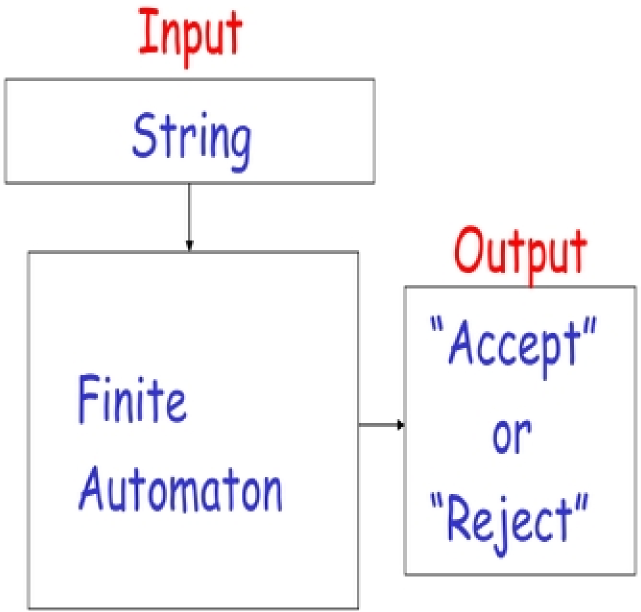
Finite Automaton



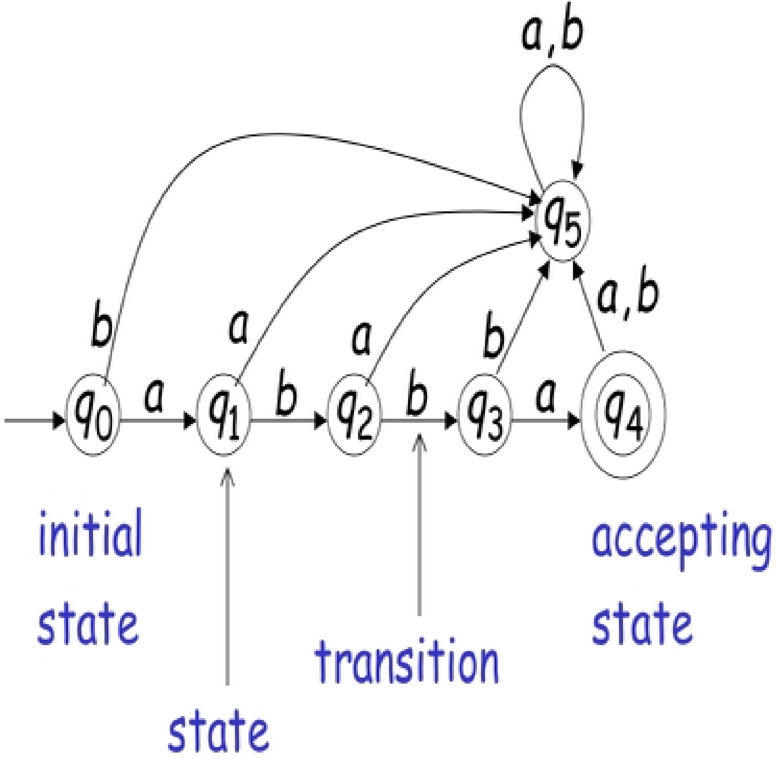
Example: Vending Machines

(small computing power)

Finite Automaton

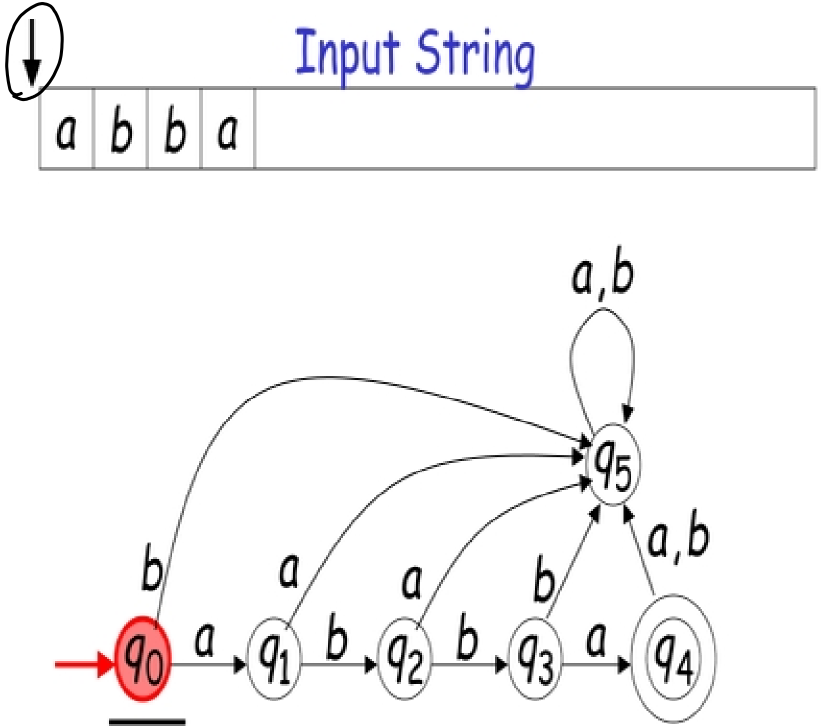


Transition Graph



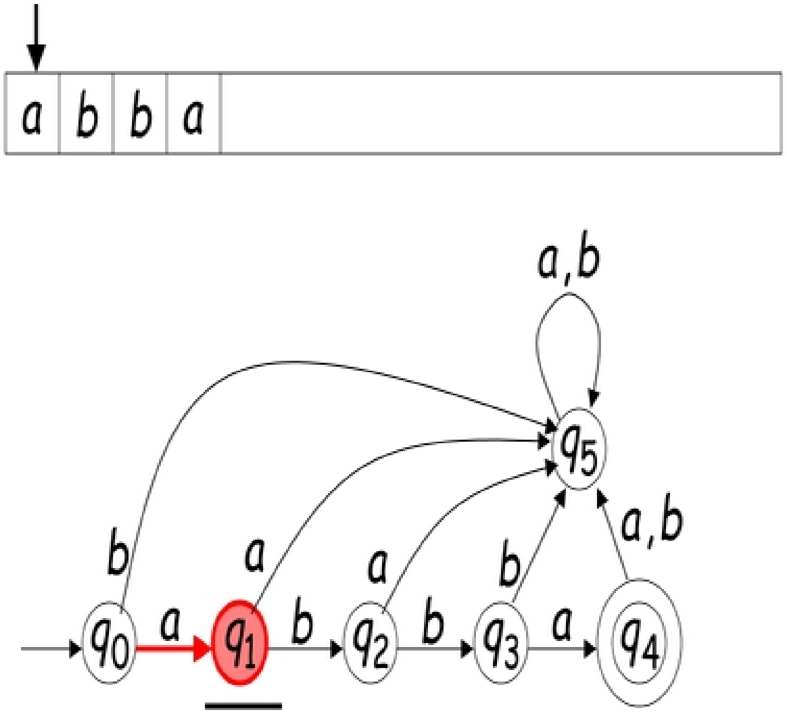
Initial Configuration

initial

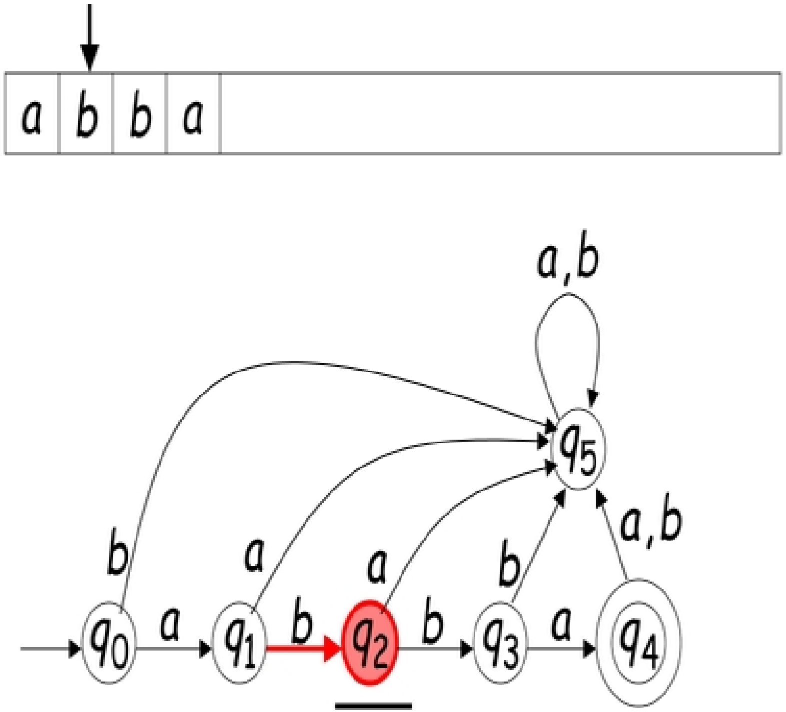


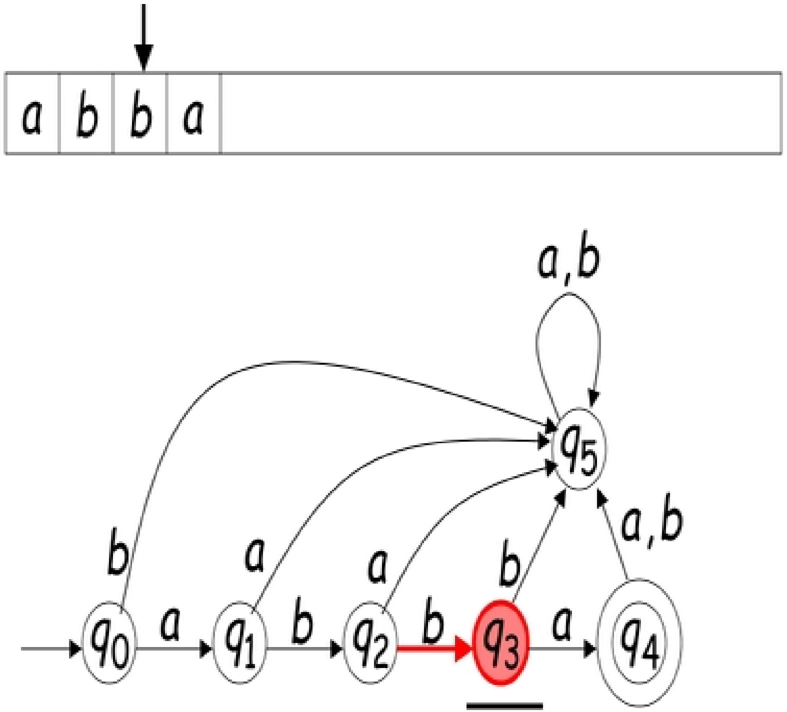
initial

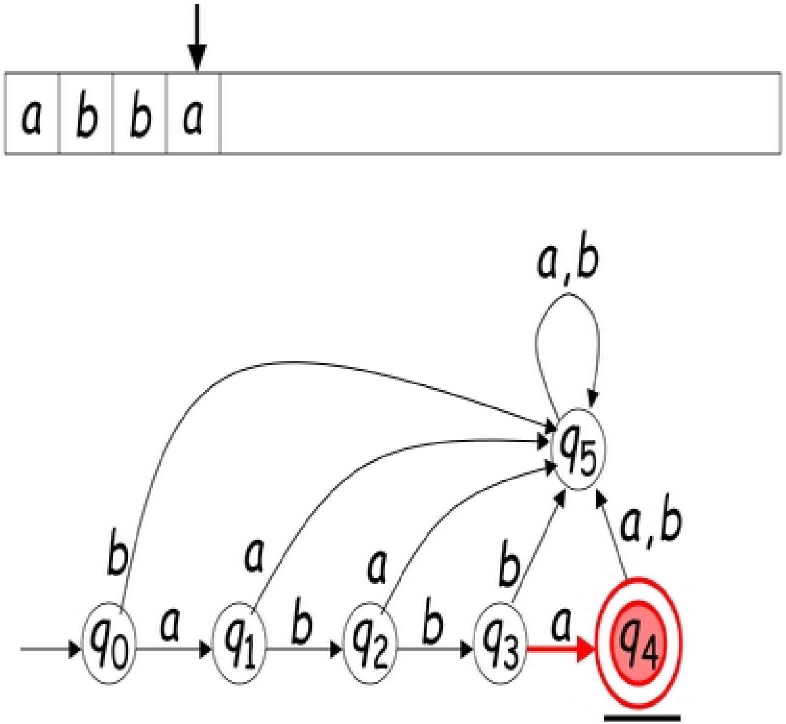
Reading the Input



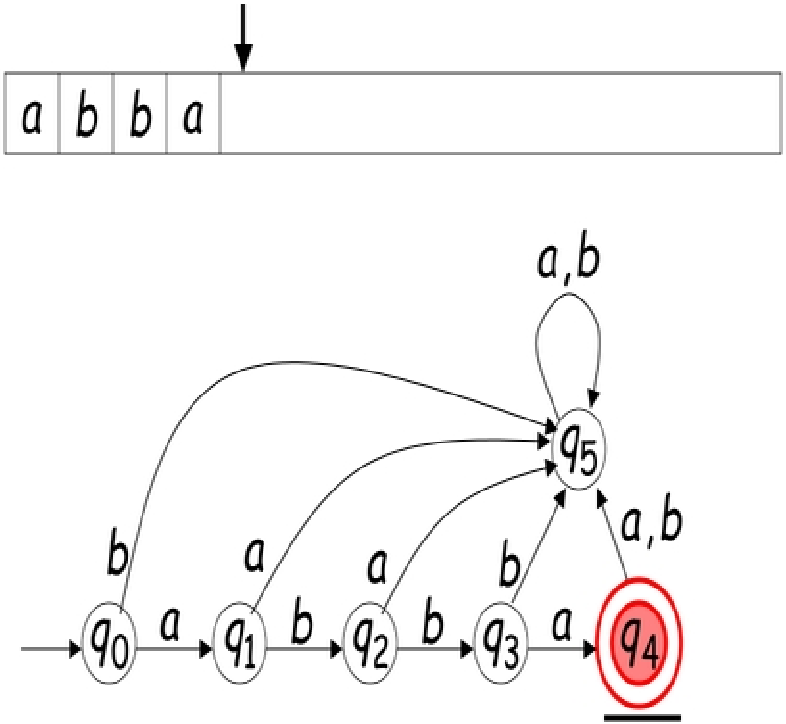
we are consuming character by character


a b b a
a,b
b
a
a
b
a,b
q0 a q1 b q2 b q3 a q4
q5


a b b a
a,b
b
a
a
b
a,b
q0 a q1 b q2 b q3 a q4
q5



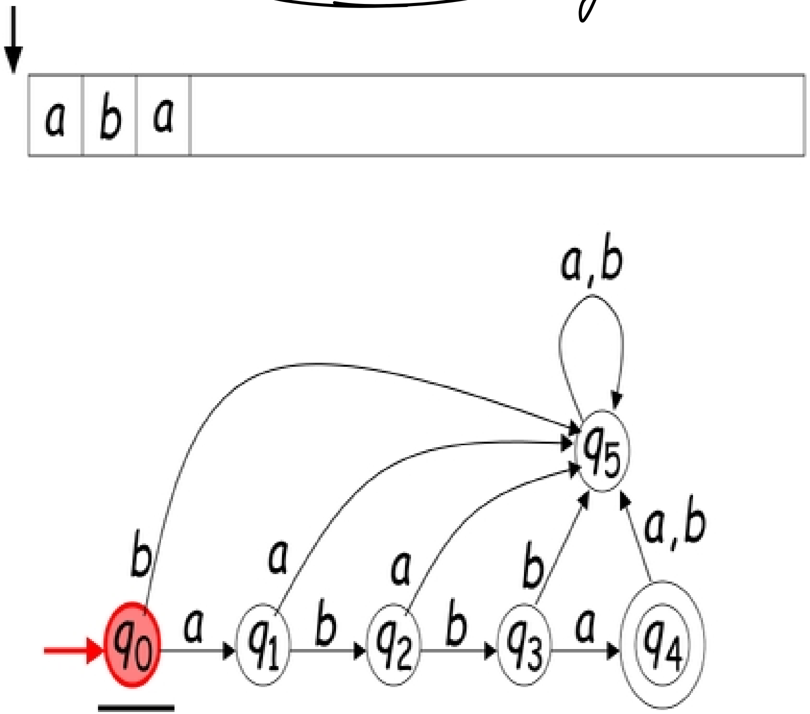
Input finished



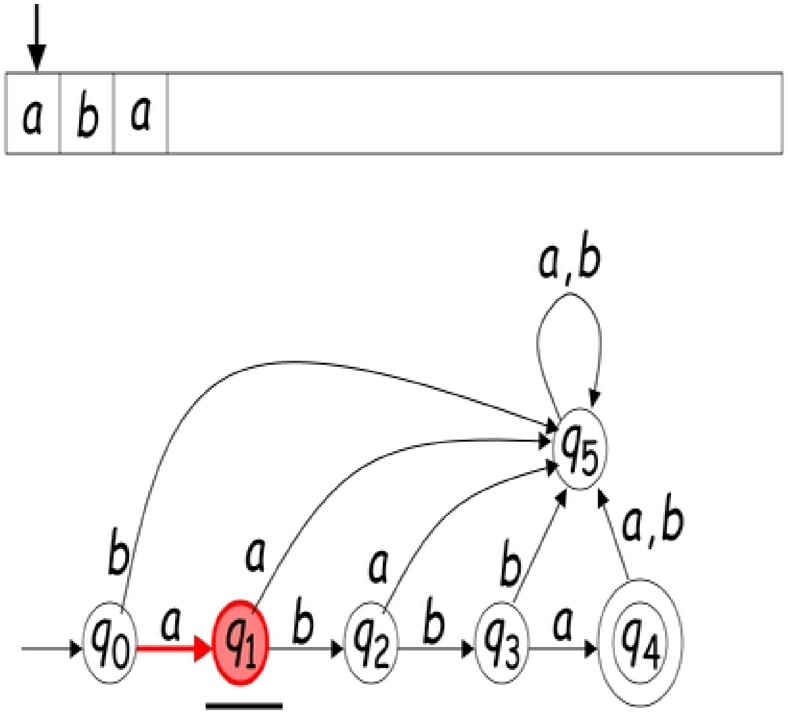
accept

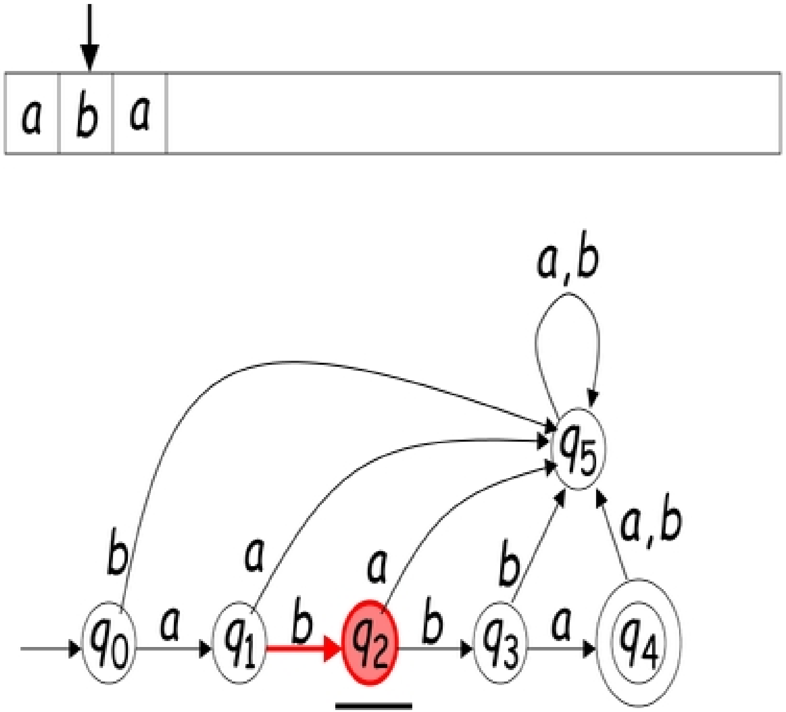
Rejection

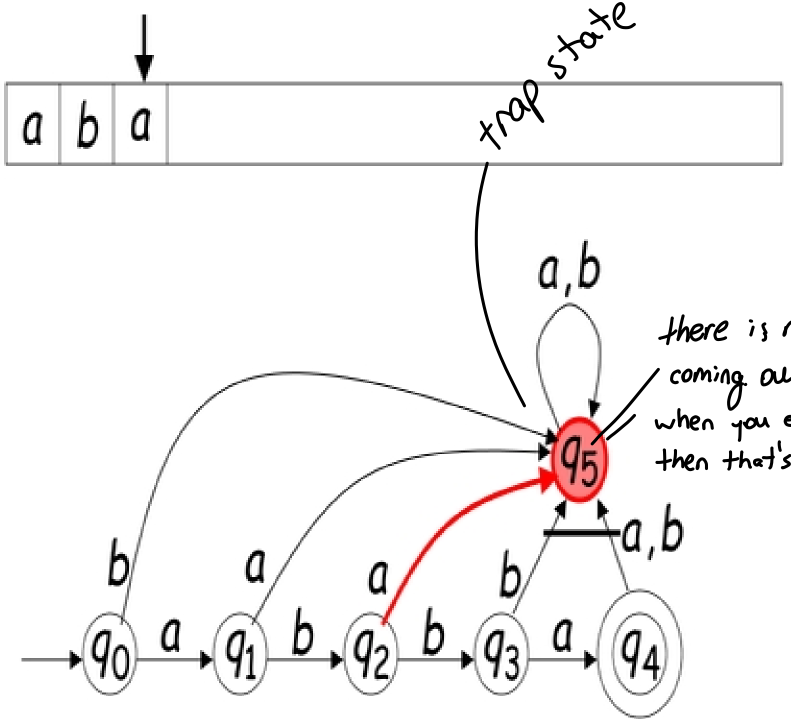
fazla olursa ve input biterse rejection



initial





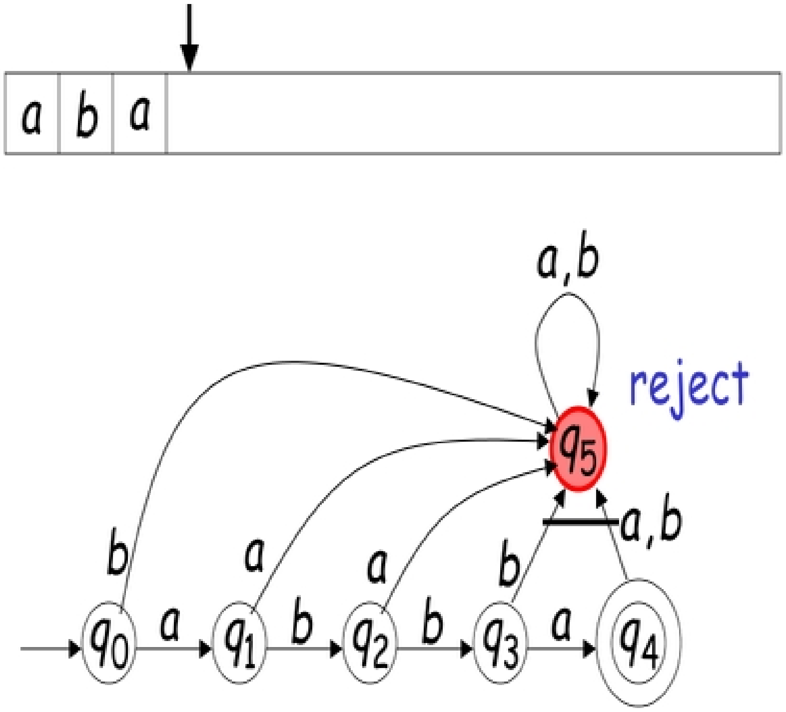


trap state

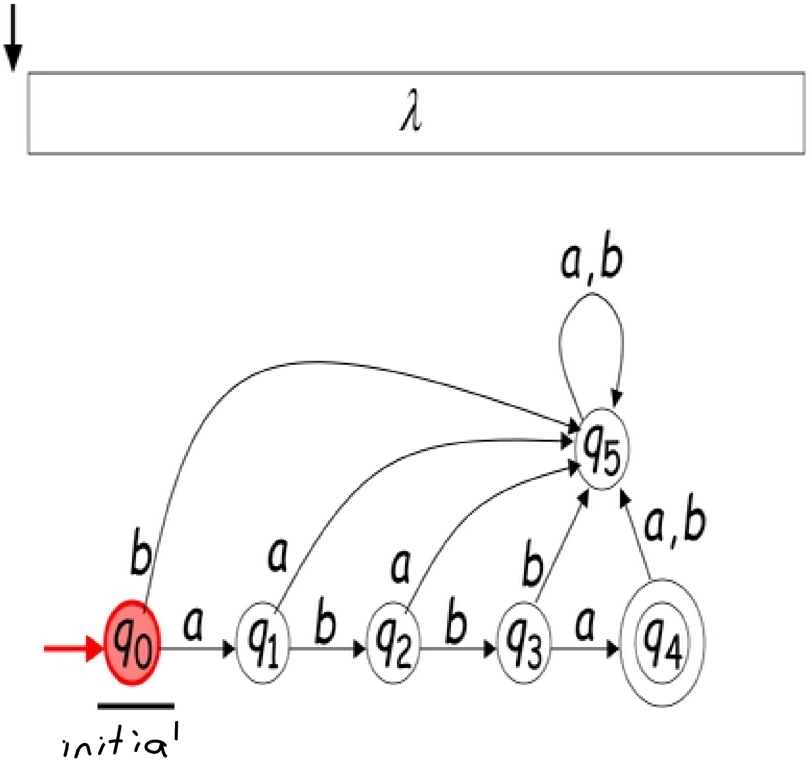
there is no coming out.

when you enter then that's all

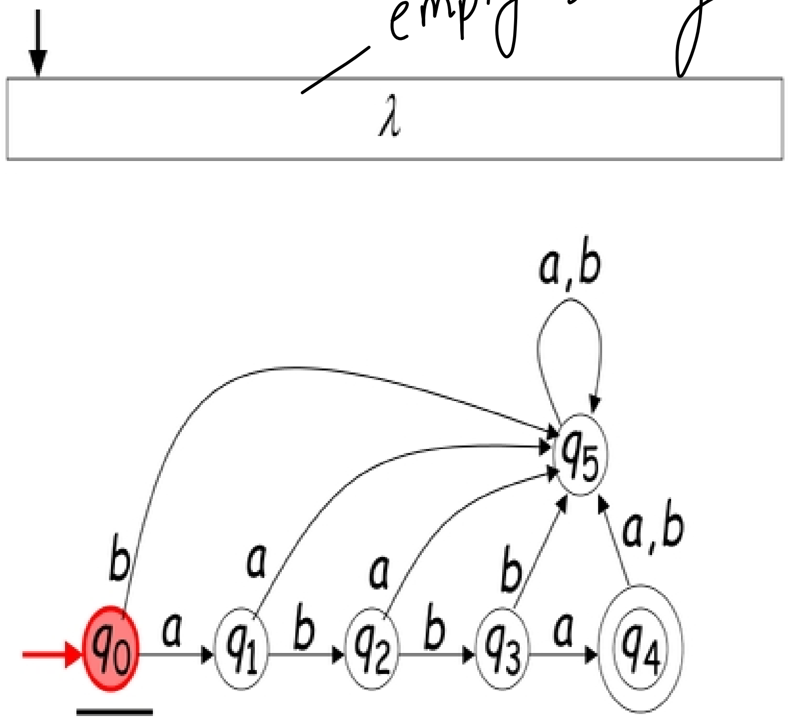
Input finished



Another Rejection



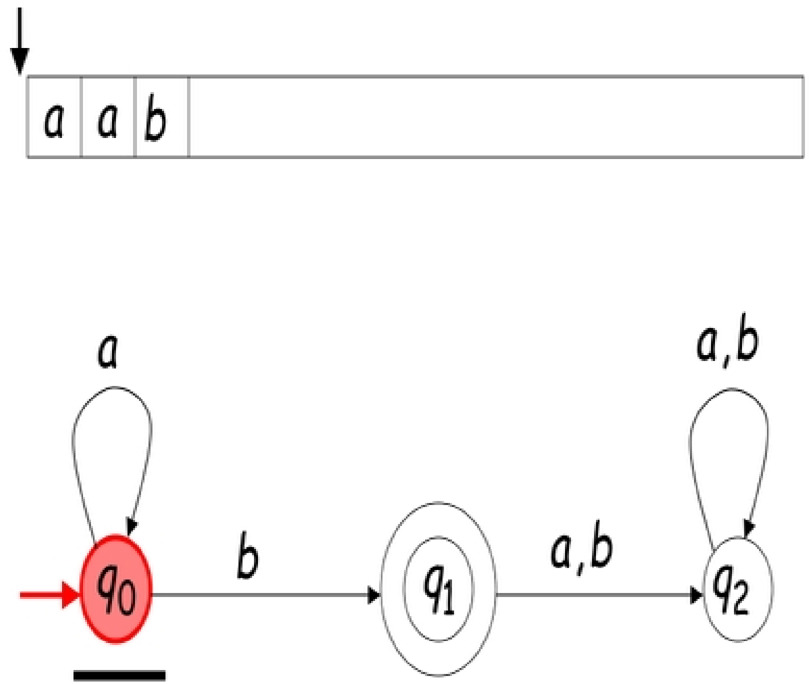
empty string

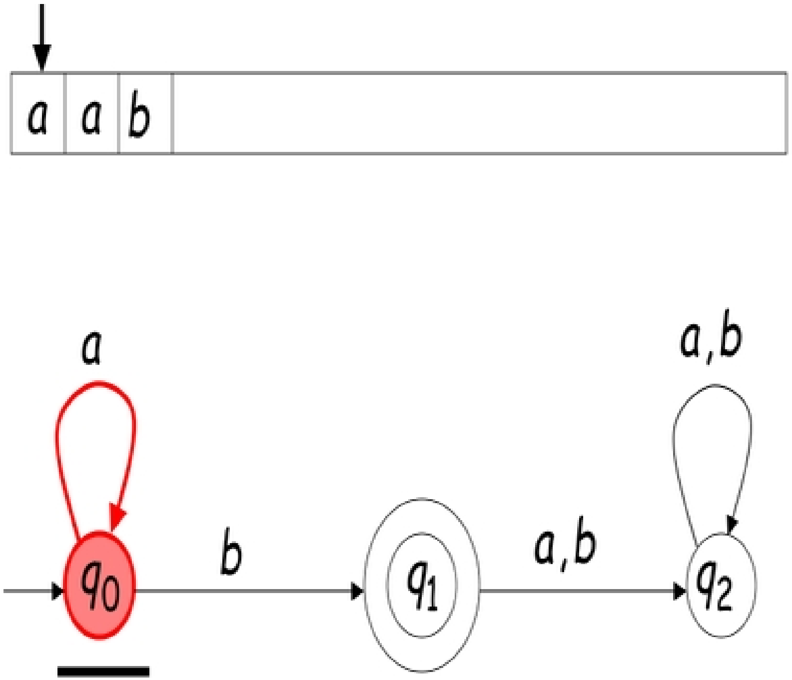


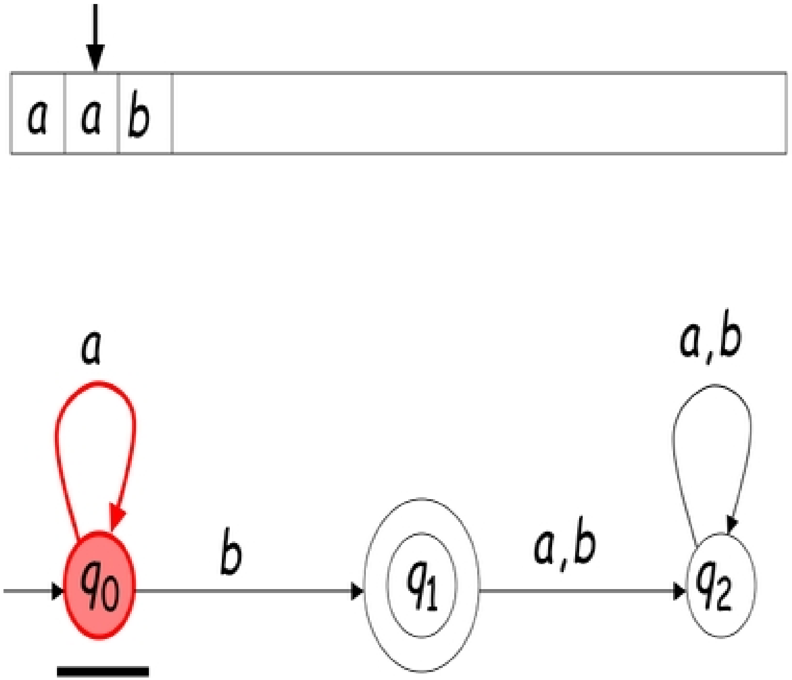
reject

automatic rejection

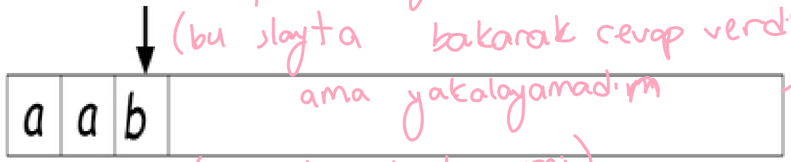
Another Example



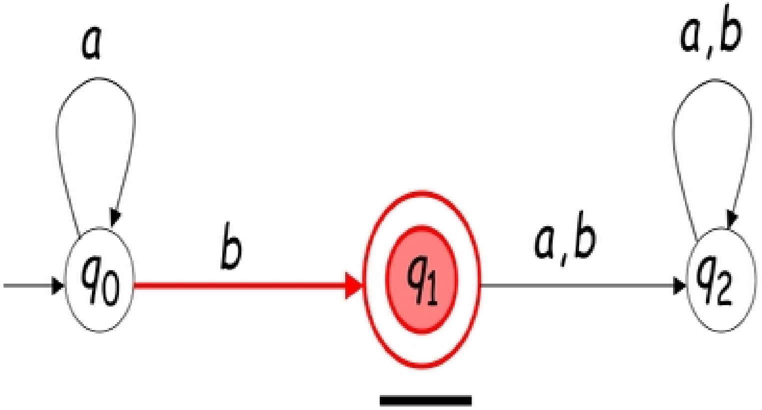

a a b
a
a,b
q_0
b
q_1
a,b
q_2


a a b
a
a,b
q_0
b
q_1
a,b
q_2

* what kind of language is this question for midterm or quiz
(bu slayta bakarak cevap verdi: ama yakalayamadım)



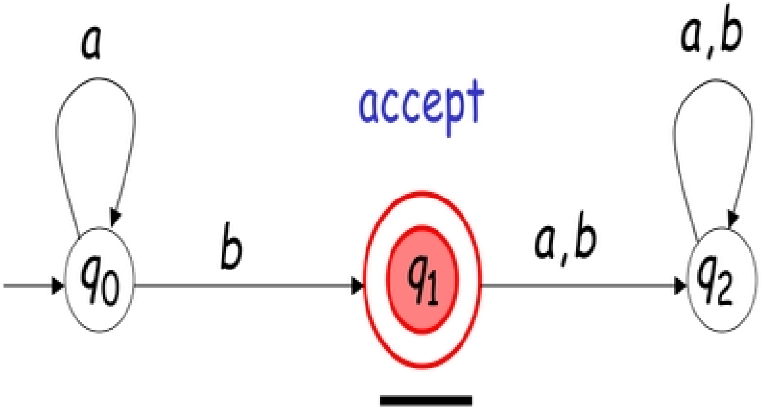
(resimden anlasılıyo yani)



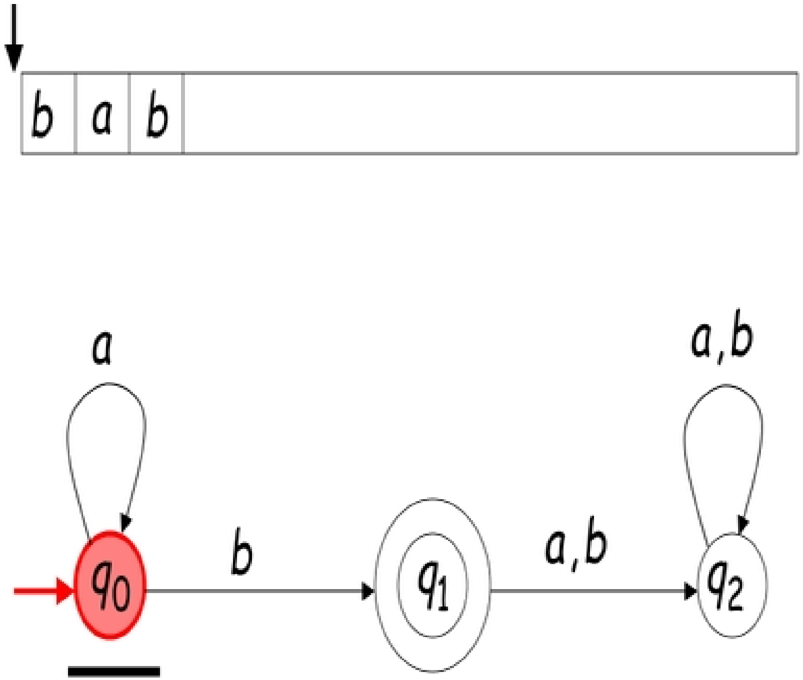
a^*b

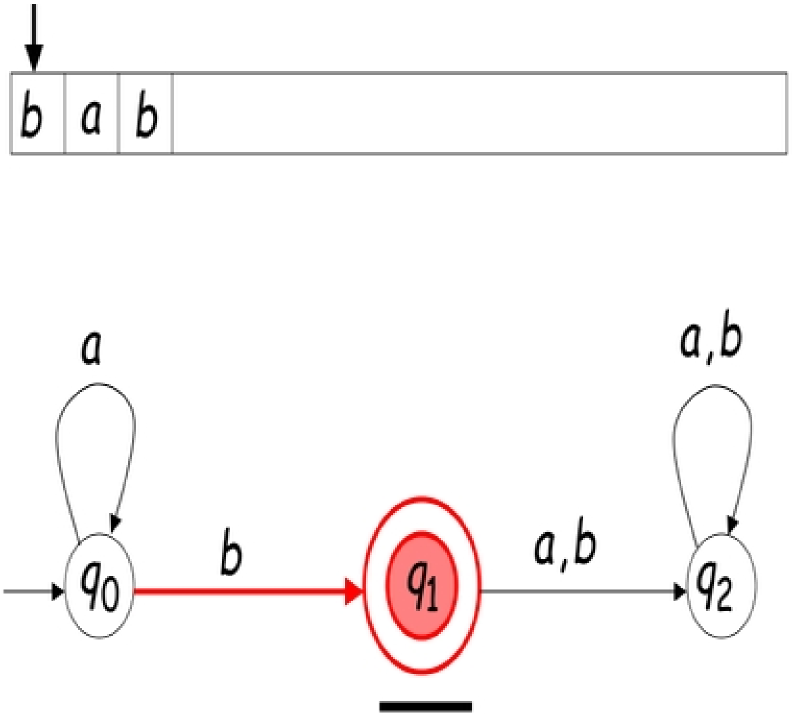
regula

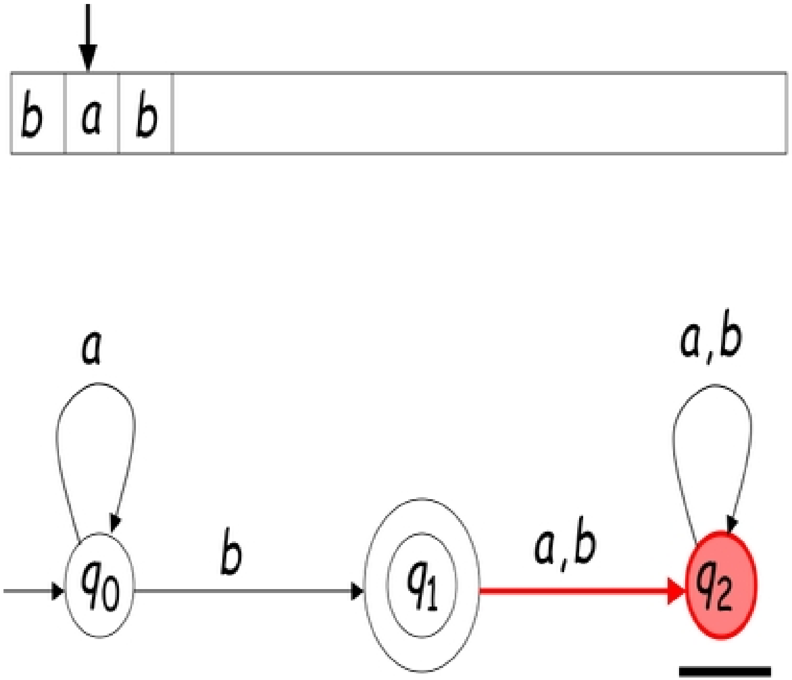
Input finished

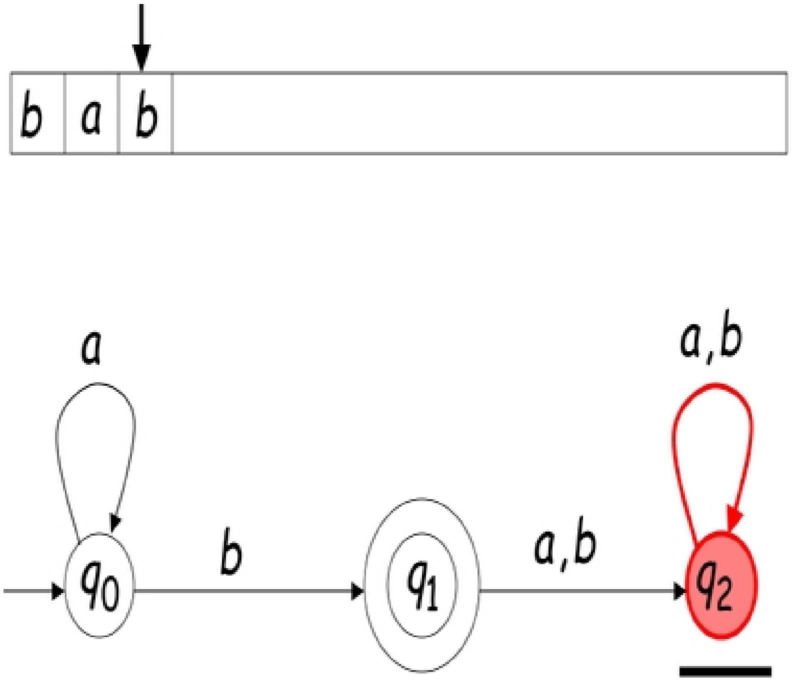


Rejection Example

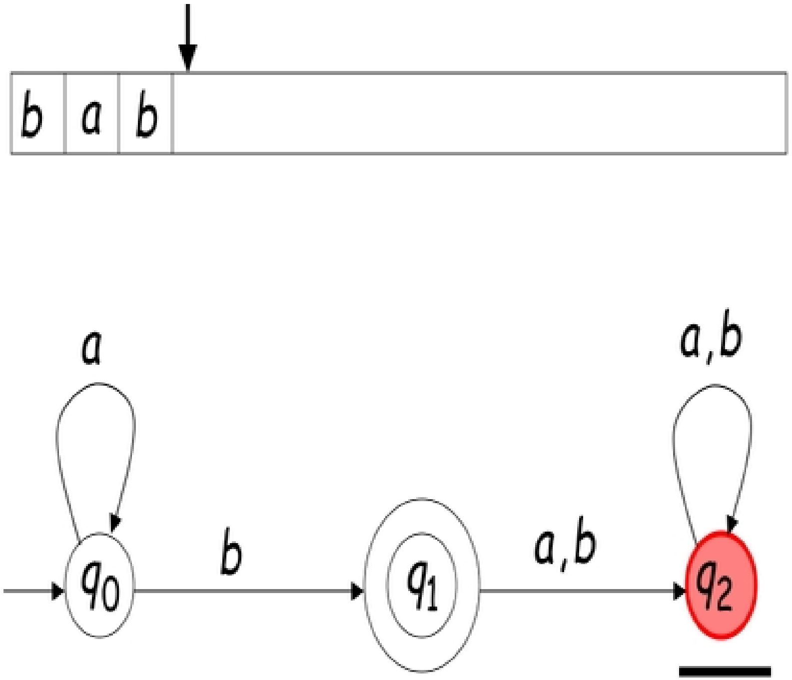





b
a
b
a
a,b
b
q_0
q_1
a,b
q_2


b
a
b
a
a,b
b
q_0
q_1
a,b
q_2

Input finished



reject

Languages Accepted by FAs

FA M

Definition:

The language $L(M)$ contains all input strings accepted by M

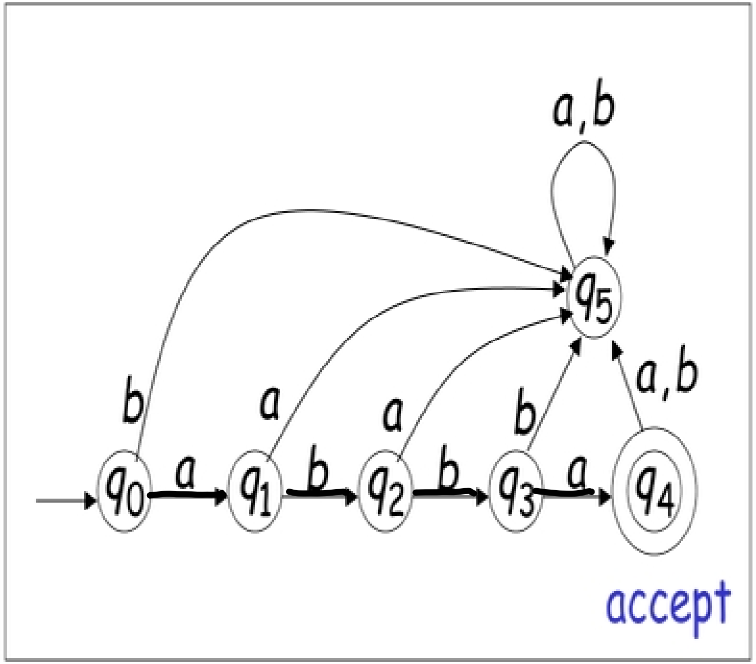
$L(M)$ = { strings that bring M to an accepting state}

Example

$L(M) = \{abba\}$

this will only accept this string

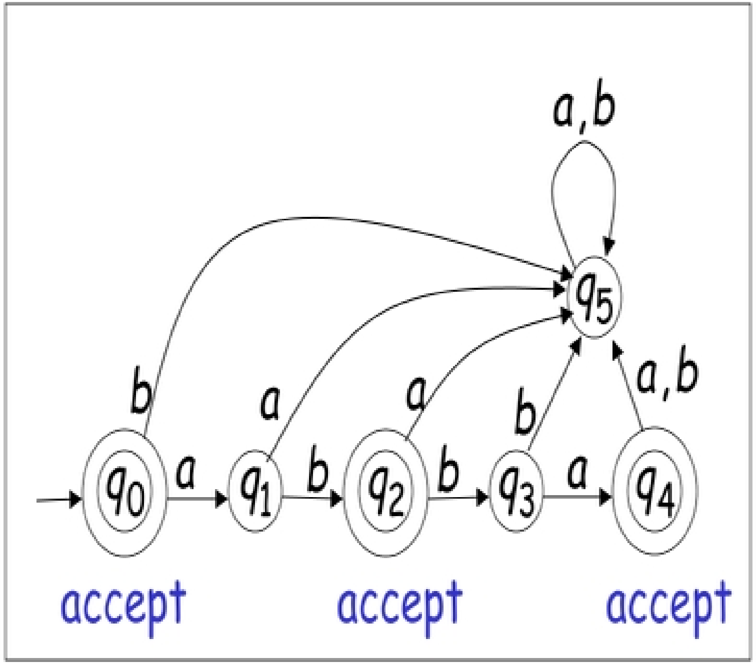
M



Example

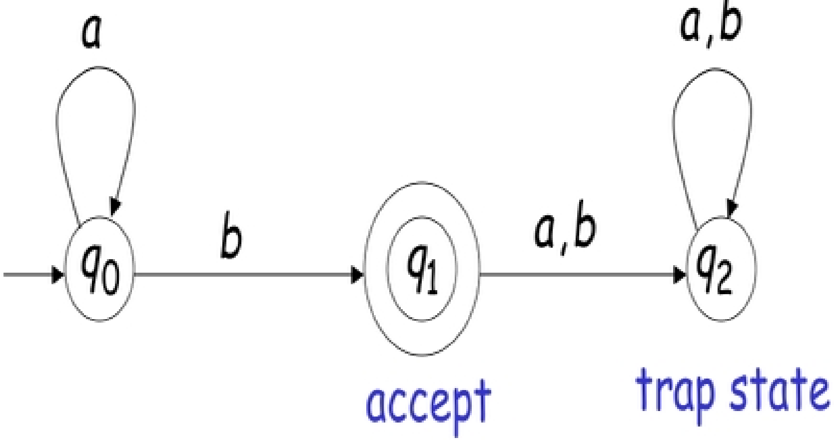
$L(M) = \{\lambda, ab, abba\}$

M



Example

$$L(M)=\{a^n b : n \geq 0\}$$



Formal Definition

Finite Automaton (FA)

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

Q : set of states

Σ : input alphabet

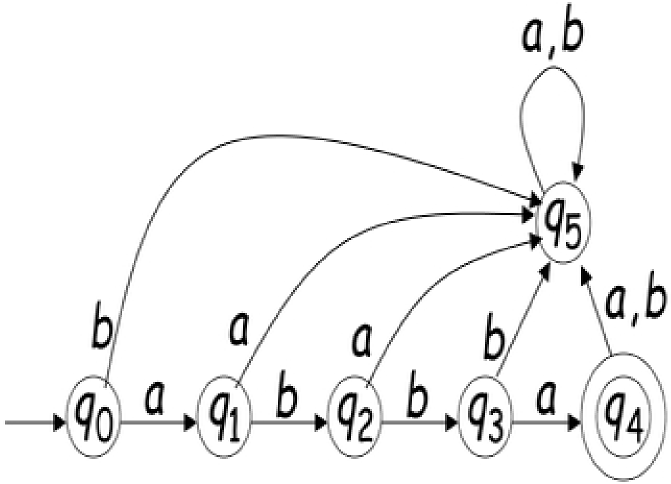
δ : transition function

q_0 : initial state

F : set of accepting states

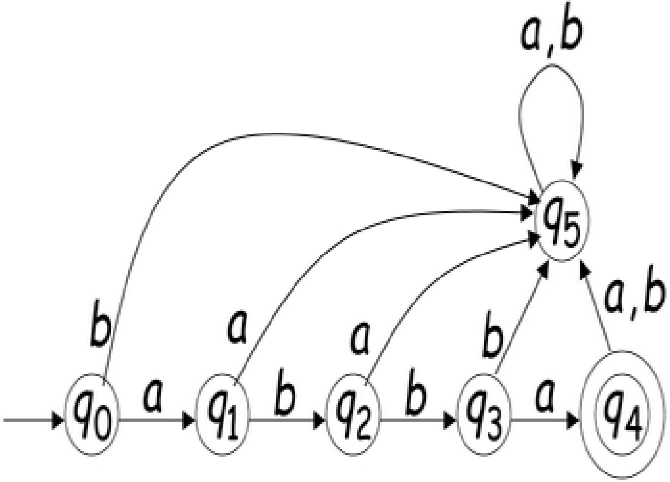
Input Alphabet Σ

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

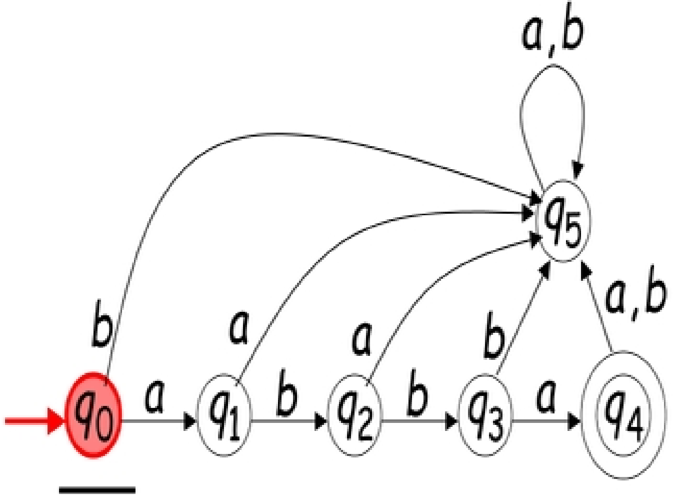


Set of States Q

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$

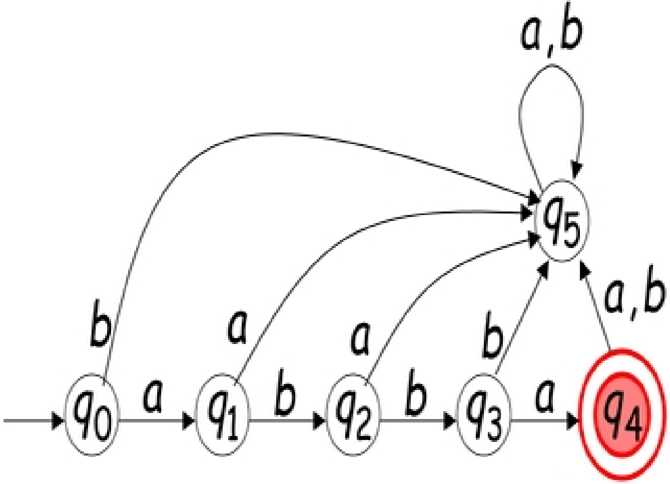


Initial State q_0



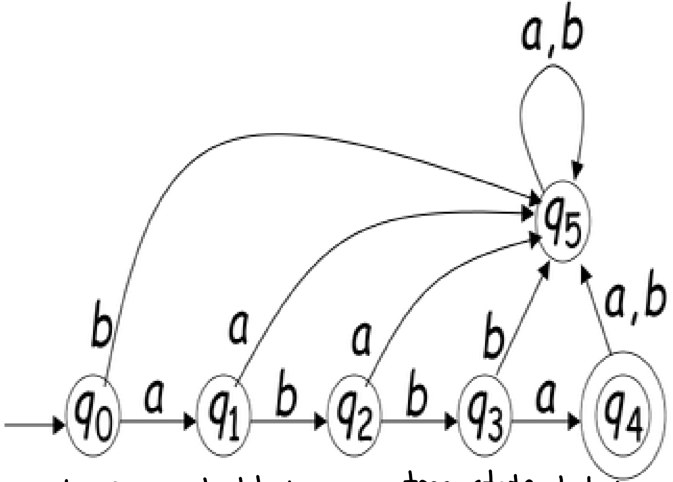
Set of Accepting States F

$$F = \{q_4\}$$



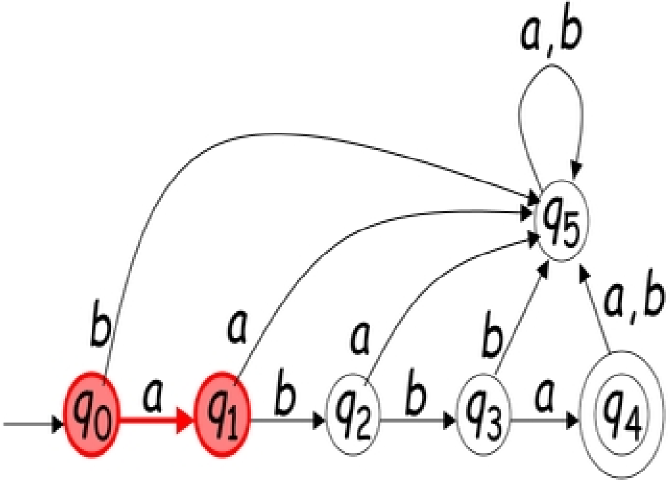
Transition Function δ

$$\delta : Q \times \Sigma \to Q$$

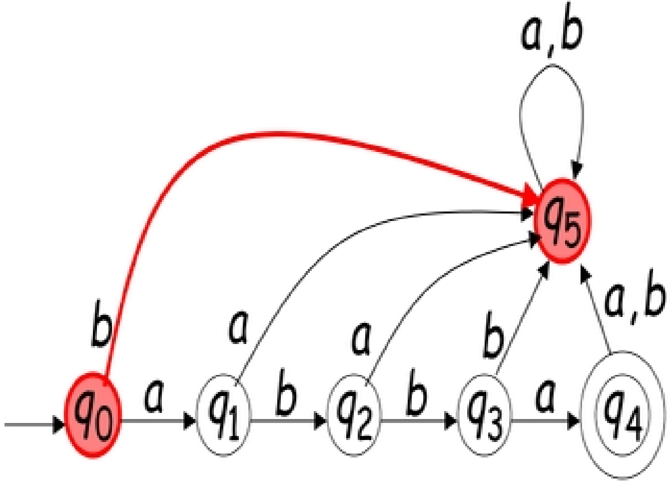


*normally all DFA should have a trap state but in practical there is not ?

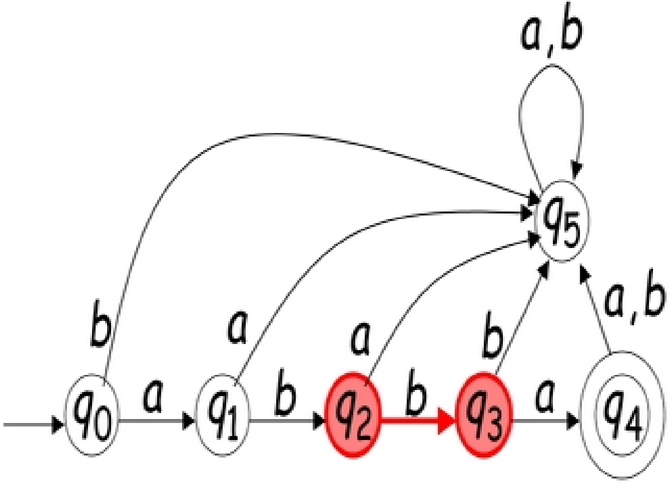
$$\delta(q_0, a) = q_1$$



$$\delta(q_0, b) = q_5$$



$$\delta(q_2, b) = q_3$$



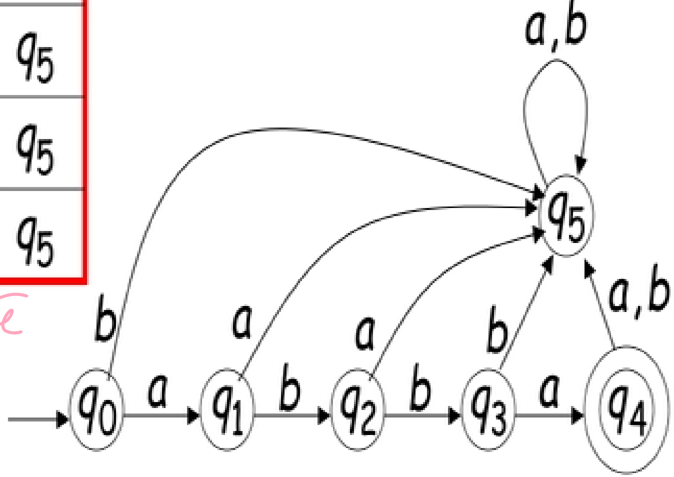
Transition Function δ

state

δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5

this is easier to check

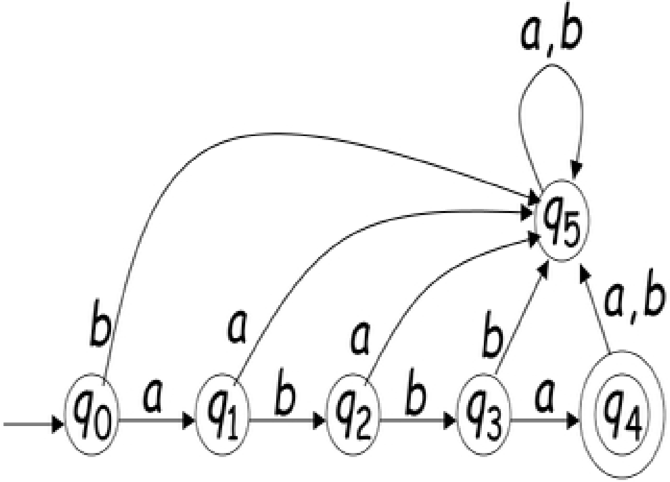
every possible transition



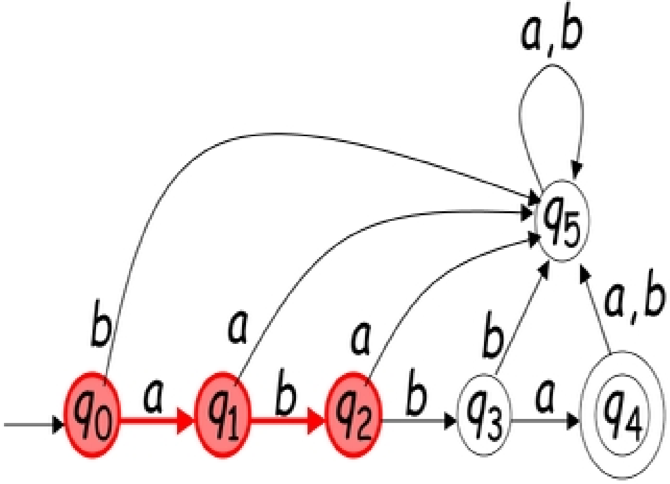
Extended Transition Function δ^*

normally δ is one, δ^* means more than 1, now you can have strings

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

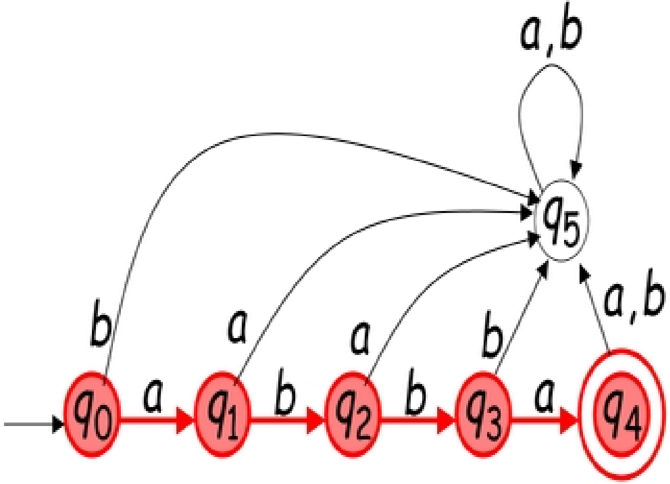


$$\delta^*(q_0, ab) = q_2$$

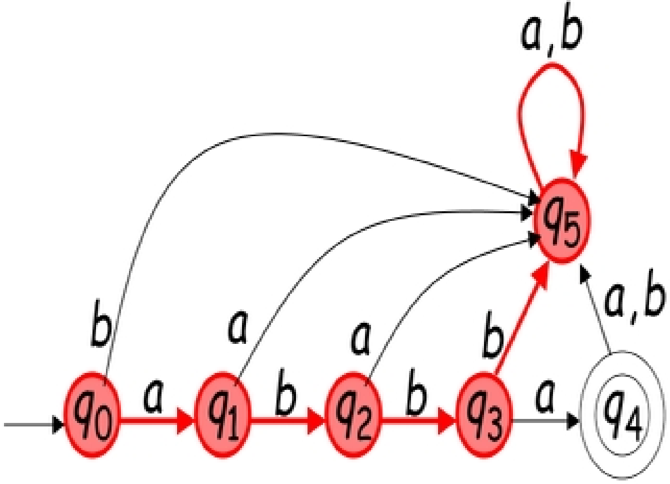


you can represent the whole thing with this transition

$$\delta^*(q_0, abba) = q_4$$



$$\delta^*(q_0, abbbaa) = q_5$$



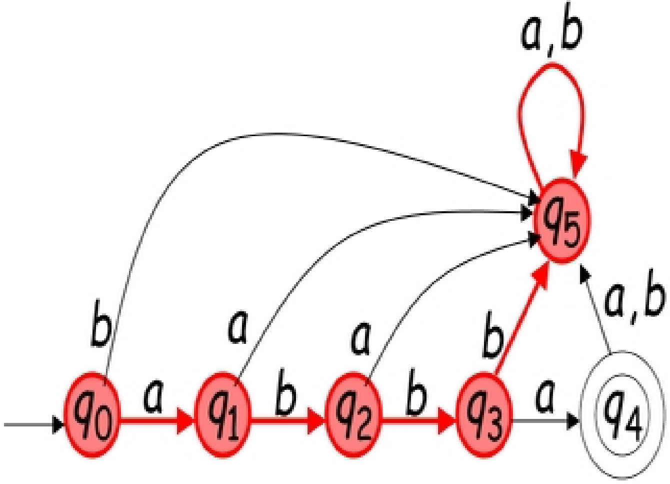
Observation: if there is a walk from q to q' with label w then

$$\delta^*(q, w) = q'$$

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

Example: There is a walk from q_0 to q_5 with label $abbbaa$

$$\delta^*(q_0, abbbaa) = q_5$$



Recursive Definition

$$\delta^*(q, \lambda) = q$$

$$\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$$

$$\left.\begin{array}{l} \delta^*(q, w\sigma) = q' \\ \delta(q_1, \sigma) = q' \end{array}\right\} \Rightarrow \delta^*(q, w\sigma) = \delta(q_1, \sigma)$$

$$\left.\begin{array}{l} \delta^*(q, w\sigma) = \delta(q_1, \sigma) \\ \delta^*(q, w) = q_1 \end{array}\right\} \Rightarrow \delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$$

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

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

q_0

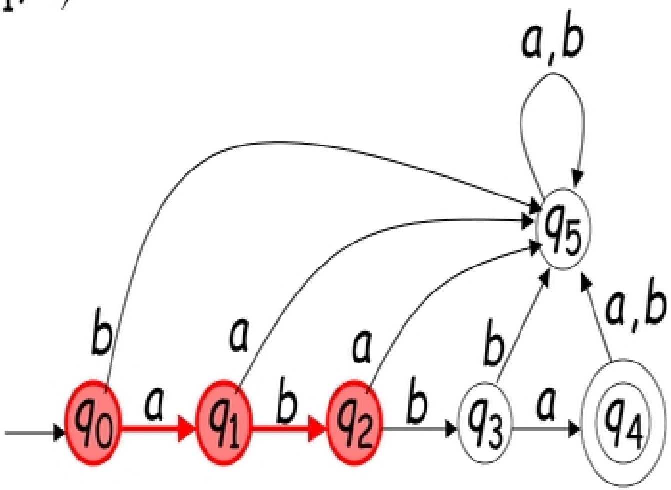
$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$

q_1

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

$\delta(q_1, b) =$ q_2

q_2



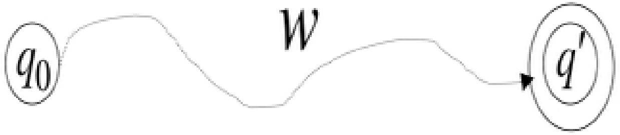
Language Accepted by FAs

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

Language accepted by M :

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

there should be a string starts initial and ends in final

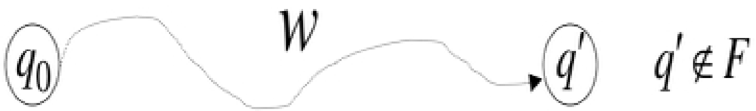


$q' \in F$

Observation

Language rejected by M :

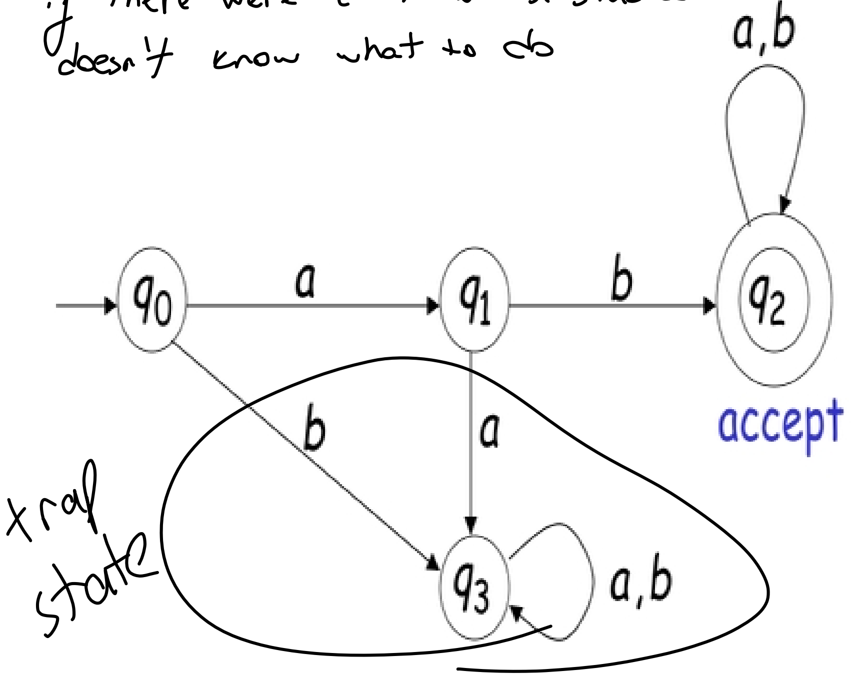
$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$



Example

$L(M) = \{$ all strings with prefix $ab\}$

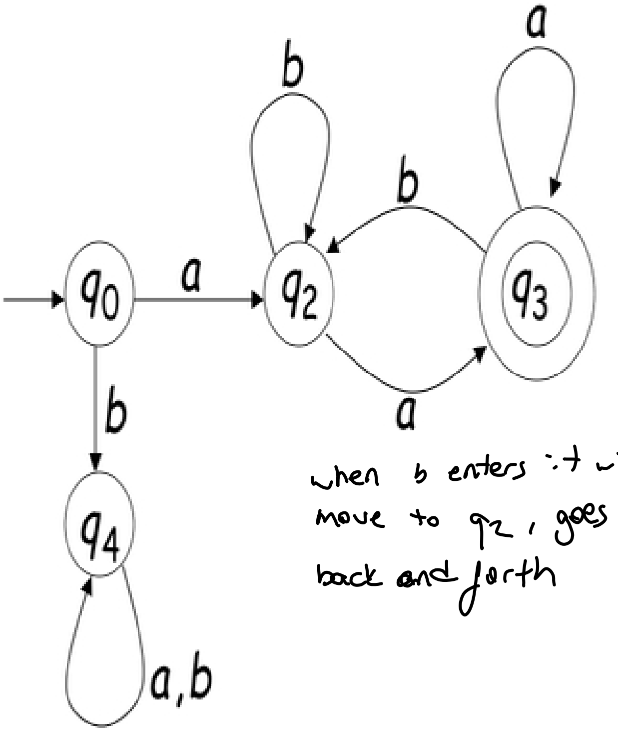
if there were "c" it will be stacked
doesn't know what to do



Example

if it keeps getting a it will be a circle

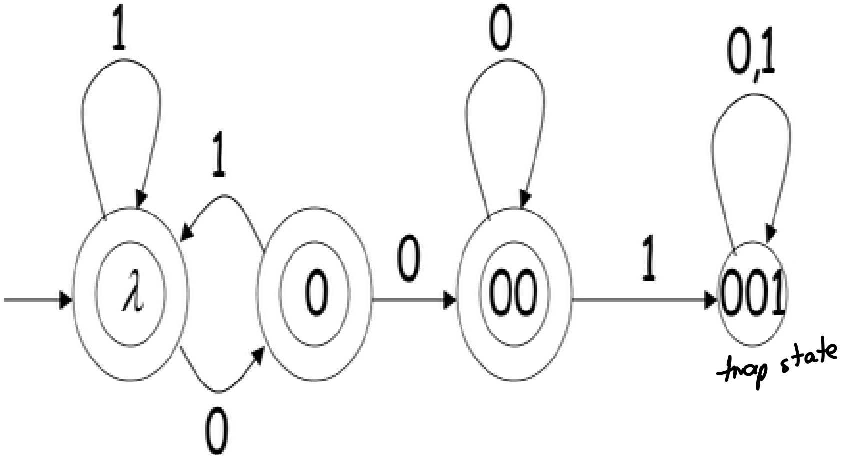
$$L(M) = \{awa : w \in \{a,b\}^*\}$$



when b enters it will move to q2, goes back and forth

Example

$L(M)$ = { all strings without substring 001 }



Regular Languages

Definition:

A language L is regular if there is FA M such that $L = L(M)$

you should be able to tell if someone asks give me a string which is regular that FA doesn't accept not

Observation:

All languages accepted by FAs form the family of regular languages

Examples of regular languages:

midterm

$\{abba\}$ $\{\lambda, ab, abba\}$

$\{awa : w \in \{a,b\}^*\}$ $\{a^n b : n \geq 0\}$

bunları hatırla

{ all strings with prefix ab }

sınavda tekrar yazmaya çalışma

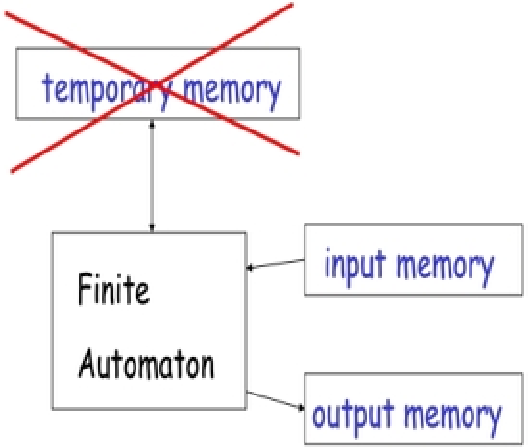
{ all strings without substring 001 }

There exist automata that accept these Languages (see previous slides).

There exist languages which are not Regular:

Example: $L = \{a^n b^n : n \geq 0\}$

There is no FA that accepts such a language



(we will prove this later in the class)