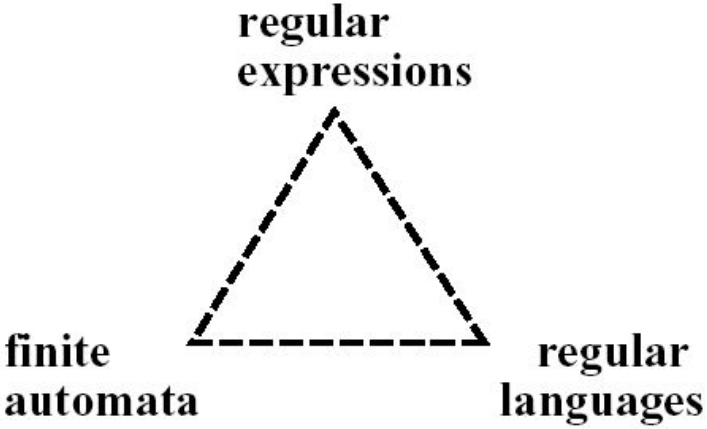
Finite State Automata

Finite-State Automata

- An RE is one way of describing a FSA.
- An RE is one way of characterizing a particular kind of formal language called a **regular language**.



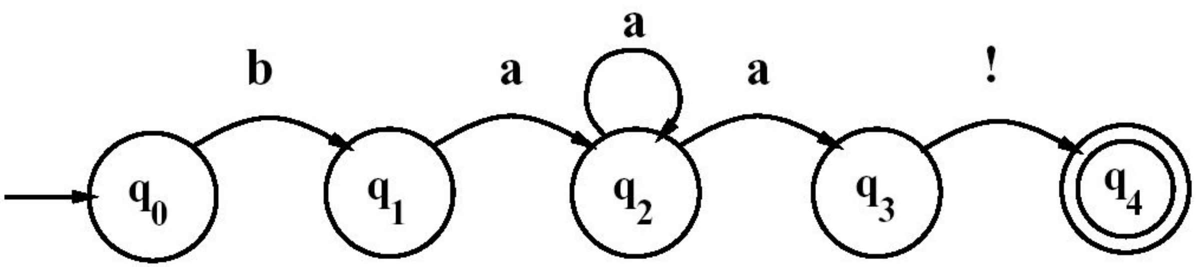
REGULAR EXPRESSION	REGULAR LANGUAGES
(a.b*)	{a, ab, abb, abbb, abbbb,….}
v*.c*	{ ε , a ,aou, aiou, b, abcd…..}

FSA

- A finite automaton is formally defined by the following five parameters:
  - Q: a finite set of N states q_0, q_1, …, q_N
  - Σ: a finite input alphabet of symbols
  - q_0: the start state
  - F: the set of final states, $F \subseteq Q$
  - $\delta(q,i)$: the transition function or transition matrix between states. Given a state $q \in Q$ and input symbol $i \in \Sigma$, $\delta(q,i)$ returns a new state $q' \in Q$. δ is thus a relation from $Q \times \Sigma$ to Q;

FSA

- **FSA** is a 5-tuple consisting of
  - **Q**: set of **states** {q0,q1,q2,q3,q4}
  - **Σ**: an **alphabet** of symbols {a,b,!}
  - **q0**: **Starting/Initial** state
  - **F**: a set of **final states** in Q {q4}
  - **δ(q,i):** a **transition function** mapping **Q x Σ to Q**



Finite-State Automata

Formal Languages

- A **formal language** is a set of strings, each string composed of symbols from a finite symbol-set call an **alphabet**.
- The usefulness of an automaton for defining a language is that it can express an infinite set in a closed form.
- A formal language may bear no resemblance at all to a real language (natural language), but
  - We often use a formal language to model part of a natural language, such as parts of the phonology, morphology, or syntax.

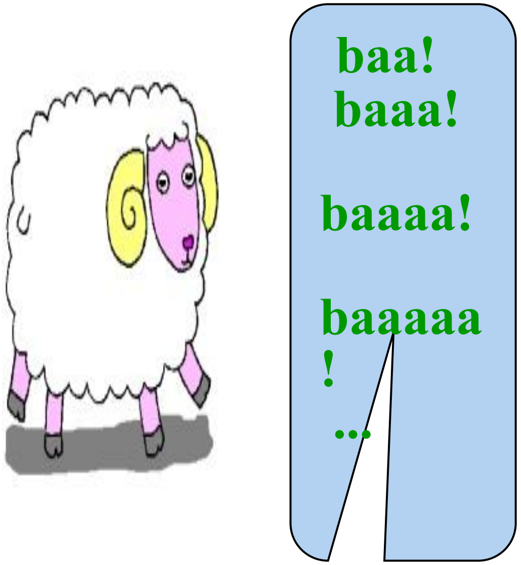
Determinism and Non-Determinism

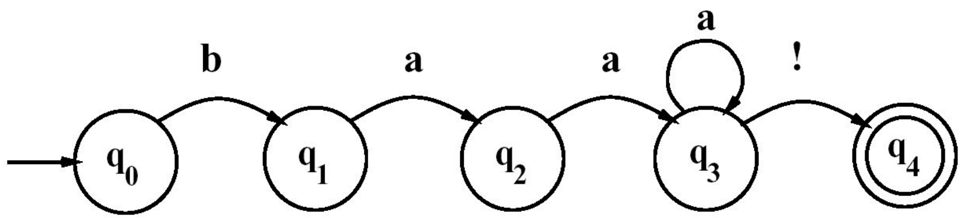
- **Deterministic**: There is at most one transition that can be taken given a current state and input symbol.

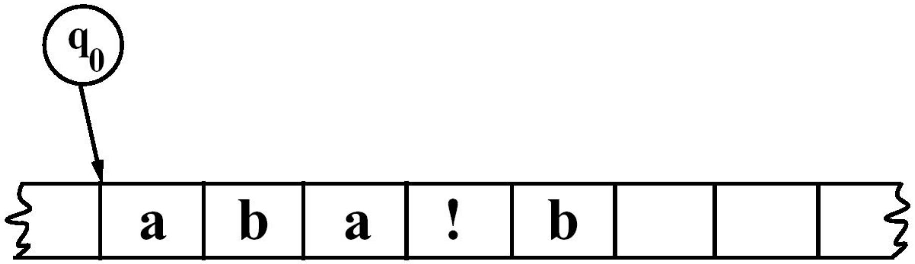
- **Non-deterministic**: There is a choice of several transitions that can be taken given a current state and input symbol. (The machine doesn’t specify how to make the choice.)

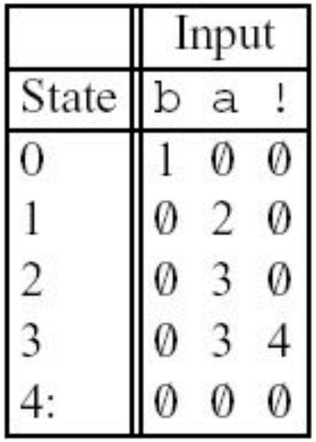
FSA:

- It recognize the regular languages represented by regular expressions
  - SheepTalk: /baa+!/





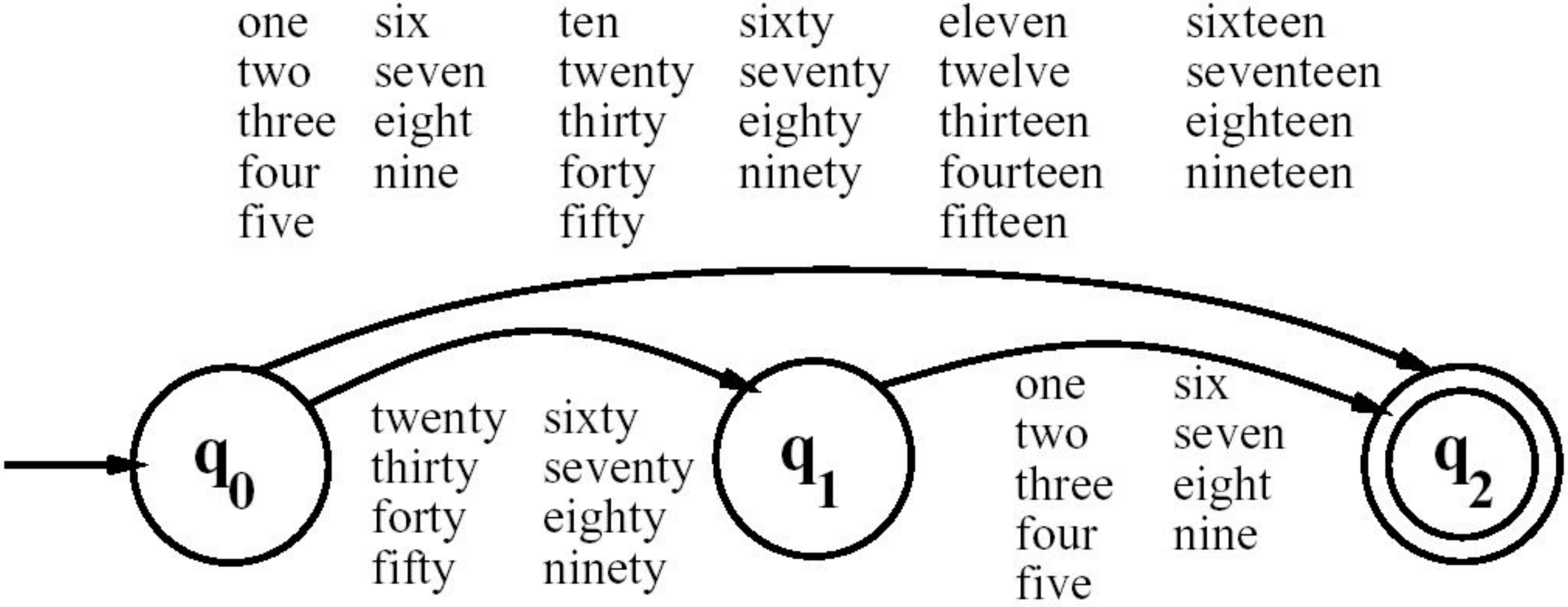




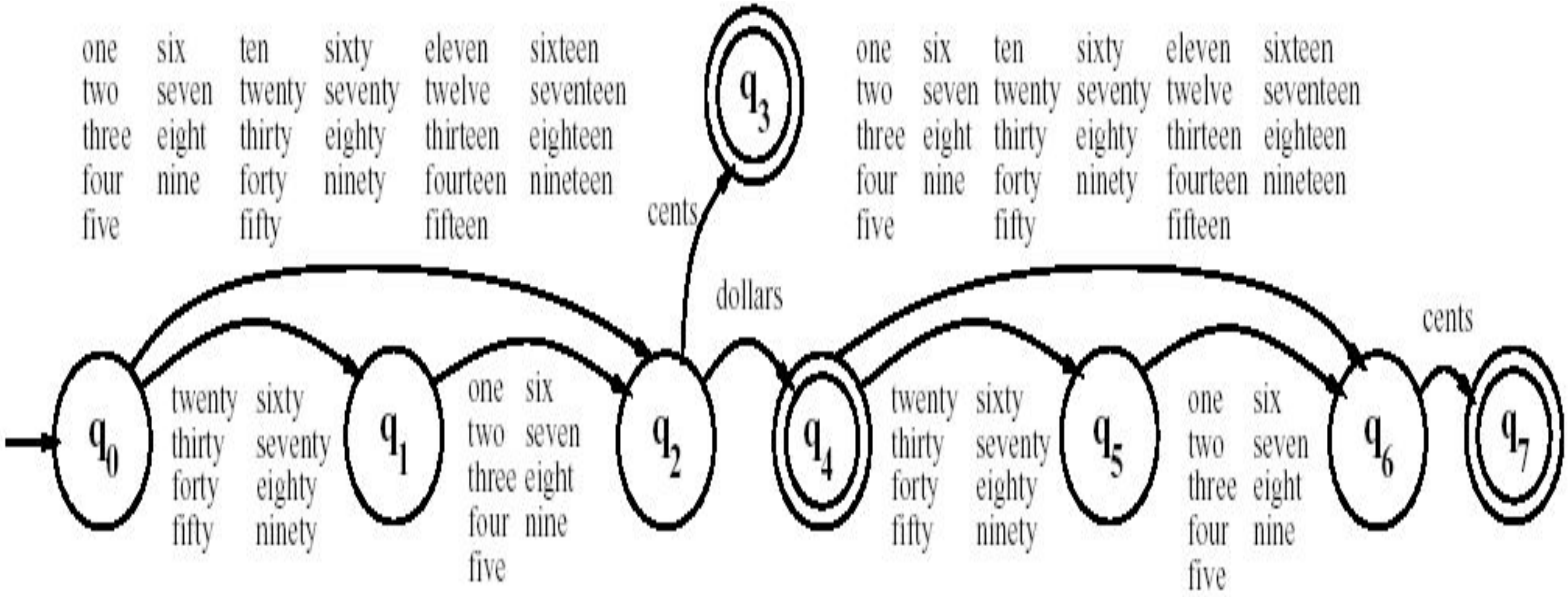
	Input		
State	b	a	!
0	1	∅	∅
1	∅	2	∅
2	∅	3	∅
3	∅	3	4
4:	∅	∅	∅

The transition-state table

FSA

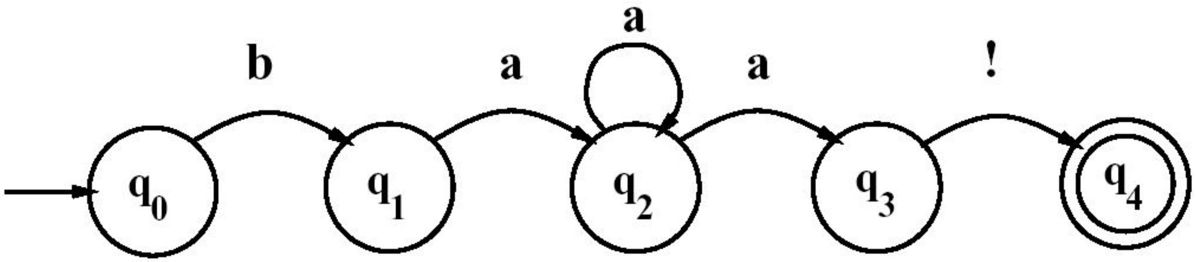

one
two
three
four
five
six
seven
eight
nine
ten
twenty
thirty
forty
fifty
sixty
seventy
eighty
ninety
eleven
twelve
thirteen
fourteen
fifteen
sixteen
seventeen
eighteen
nineteen
q0
twenty
thirty
forty
fifty
sixty
seventy
eighty
ninety
q1
one
two
three
four
five
six
seven
eight
nine
q2

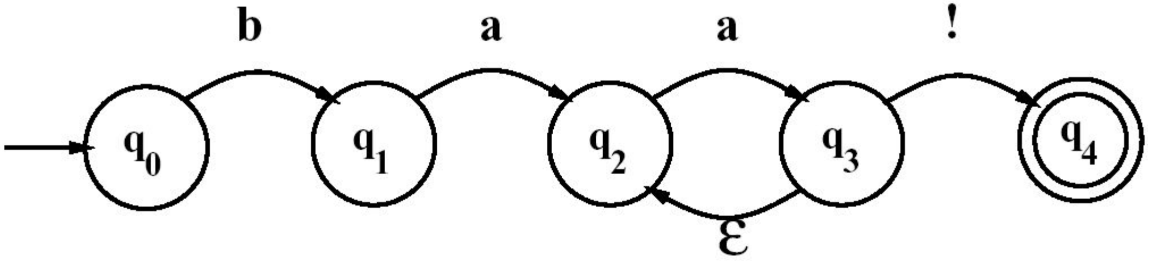
FSA



one two three four five six seven eight nine ten twenty thirty forty fifty sixty seventy eighty ninety eleven twelve thirteen fourteen fifteen sixteen seventeen eighteen nineteen
q3
cents
dollars
one two three four five six seven eight nine ten twenty thirty forty fifty sixty seventy eighty ninety eleven twelve thirteen fourteen fifteen sixteen seventeen eighteen nineteen
cents
q0
twenty thirty forty fifty sixty seventy eighty ninety
q1
one two three four five six seven eight nine
q2
q4
twenty thirty forty fifty sixty seventy eighty ninety
q5
one two three four five six seven eight nine
q6
q7

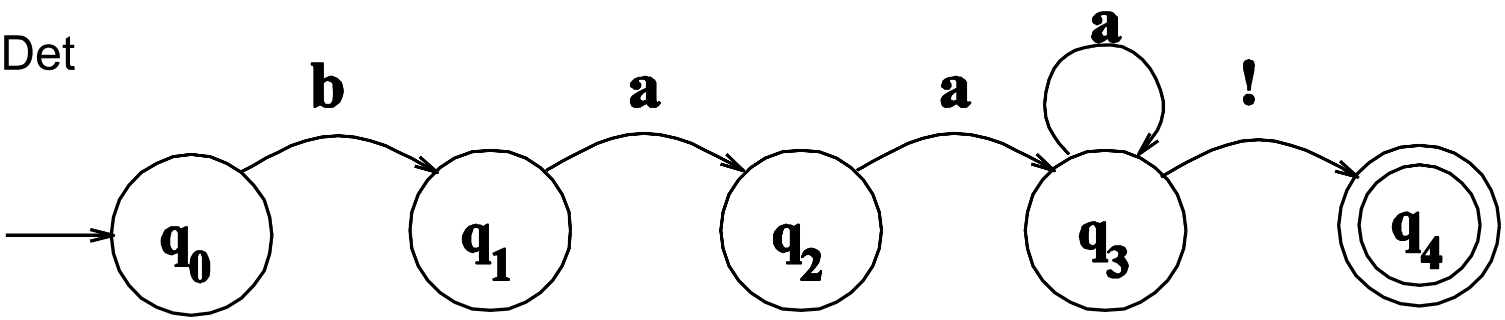
FSA: Non-Deterministic FSAs



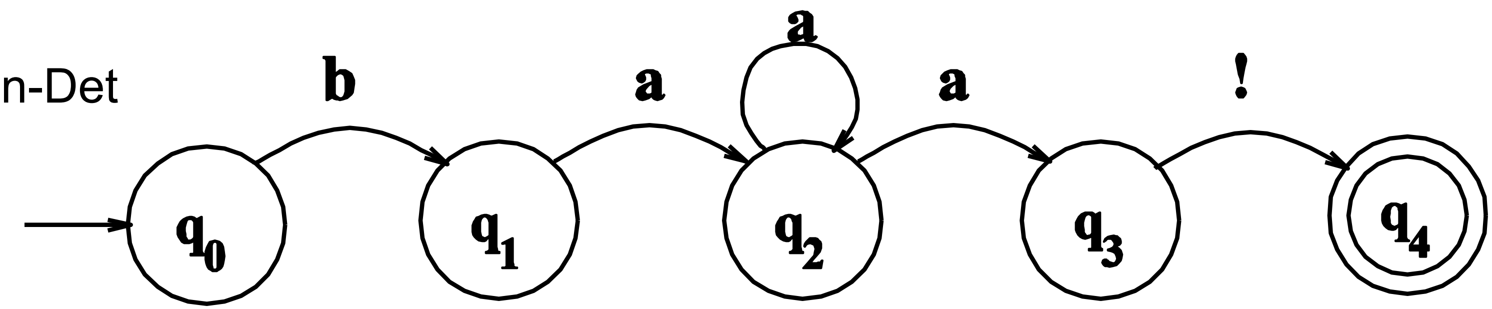


NFSA

Det



Non-Det

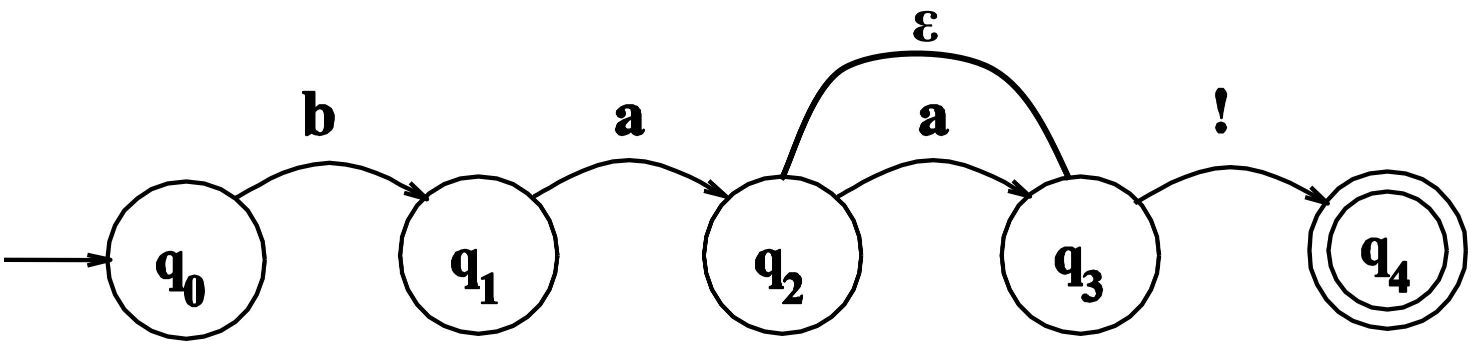


NFSA

		Input			
State		b	a	!	ε
0		1	Ø	Ø	Ø
1		Ø	2	Ø	Ø
2		Ø	2,3	Ø	Ø
3		Ø	Ø	4	Ø
④		Ø	Ø	Ø	Ø

NFSA

- Another technique
  - Epsilon transitions
    - these transitions do not examine or advance the tape during recognition



Formalities

- Deterministic Finite Accepter (DFA)

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

Q : set of states

Σ : input alphabet

δ : transition function

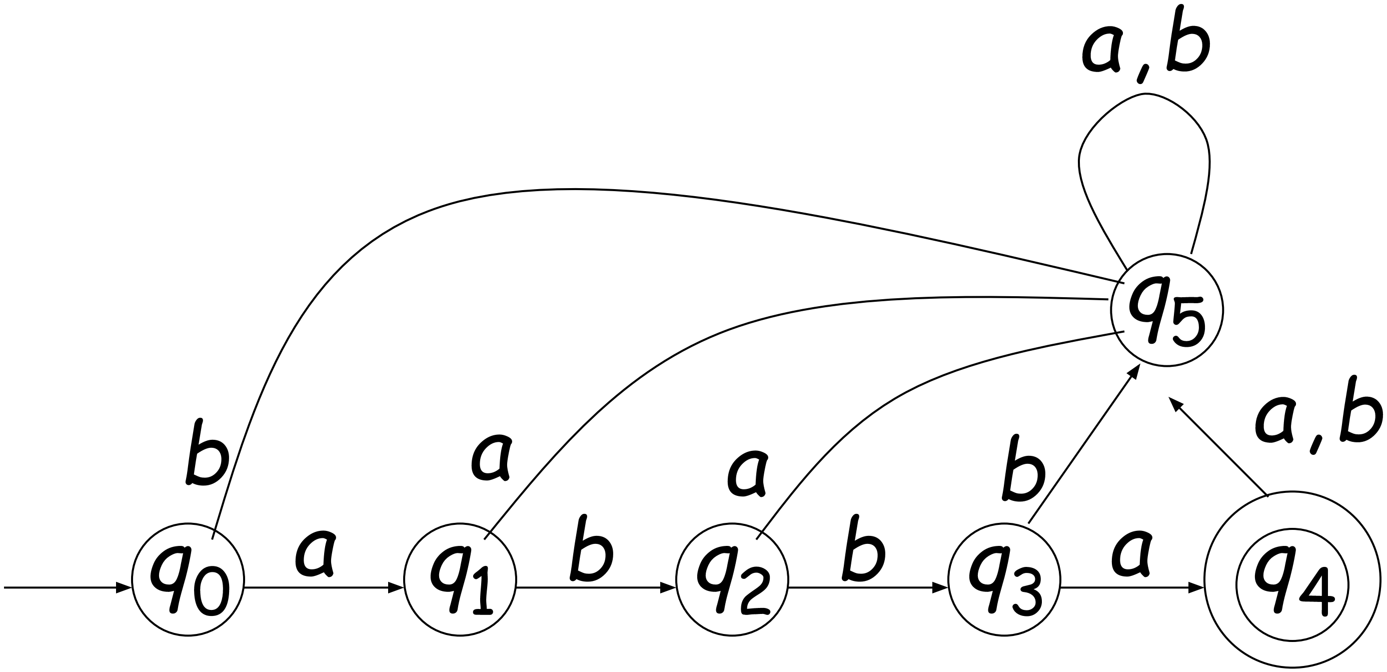
q_0 : initial state

F : set of final states

Input Aplhabet

Σ

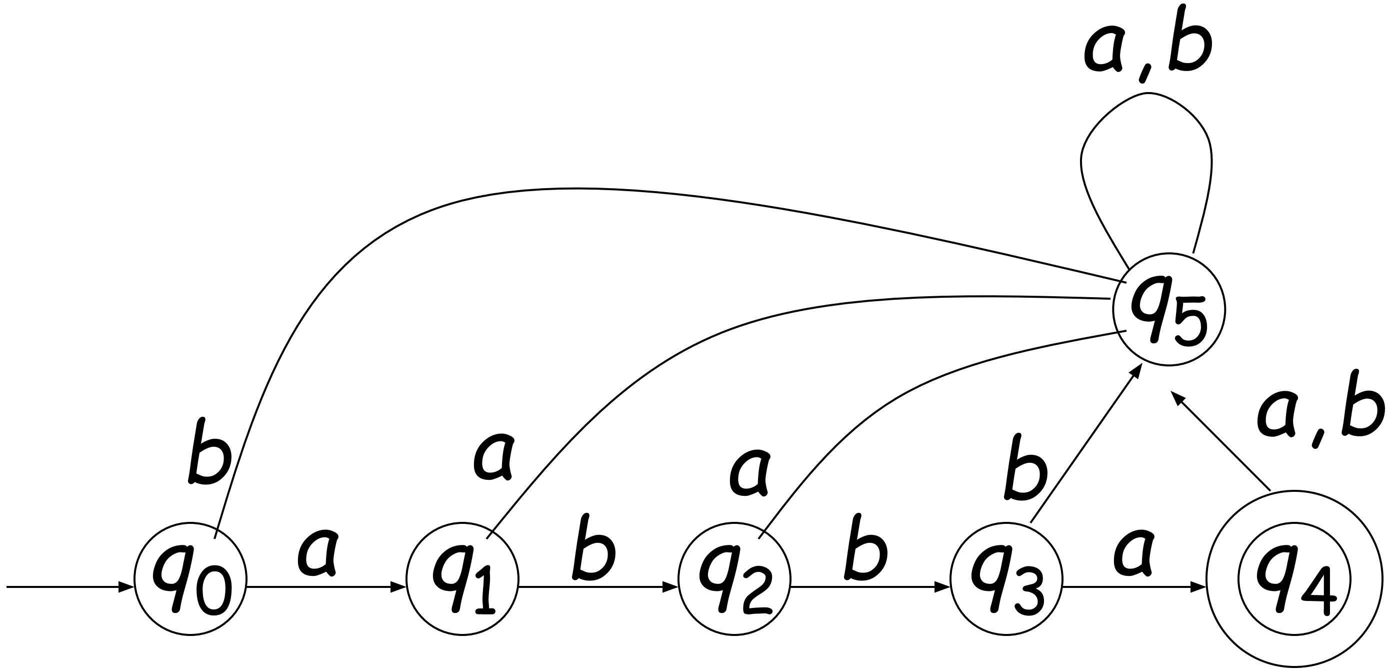
- $\Sigma = \{a, b\}$



Set of States

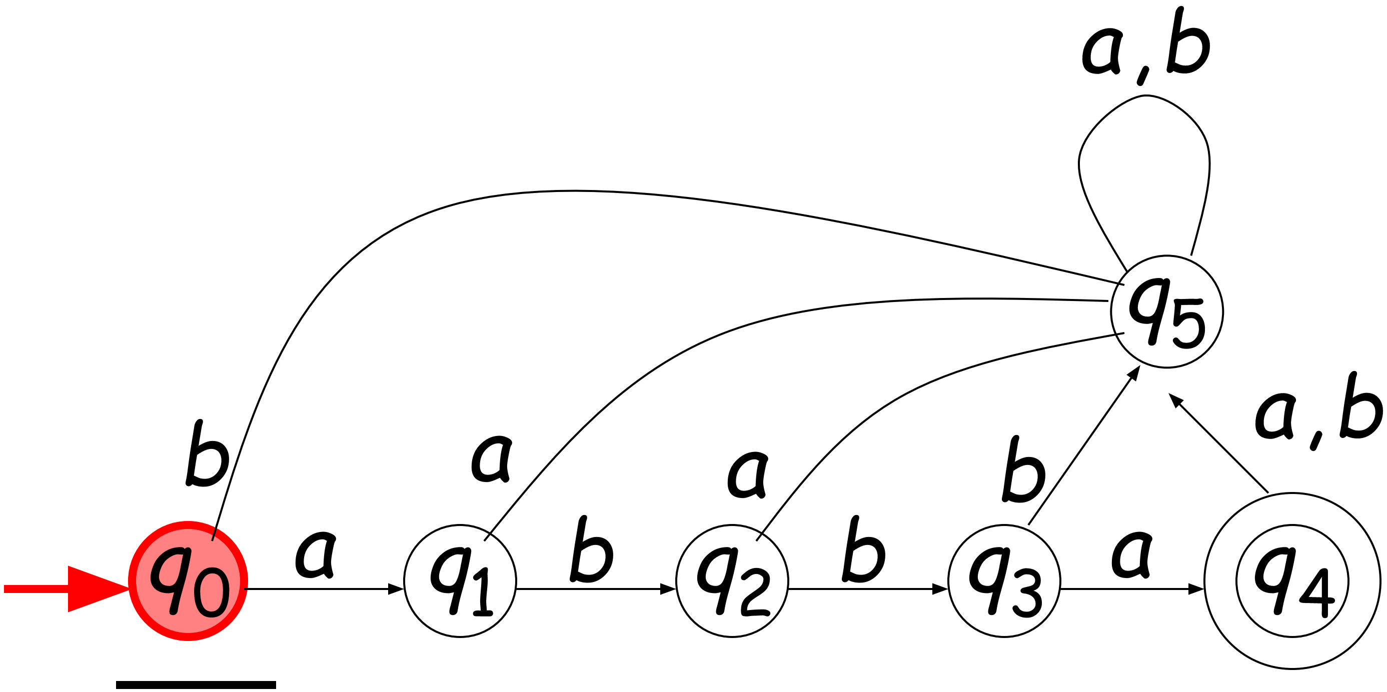
Q

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



Initial State

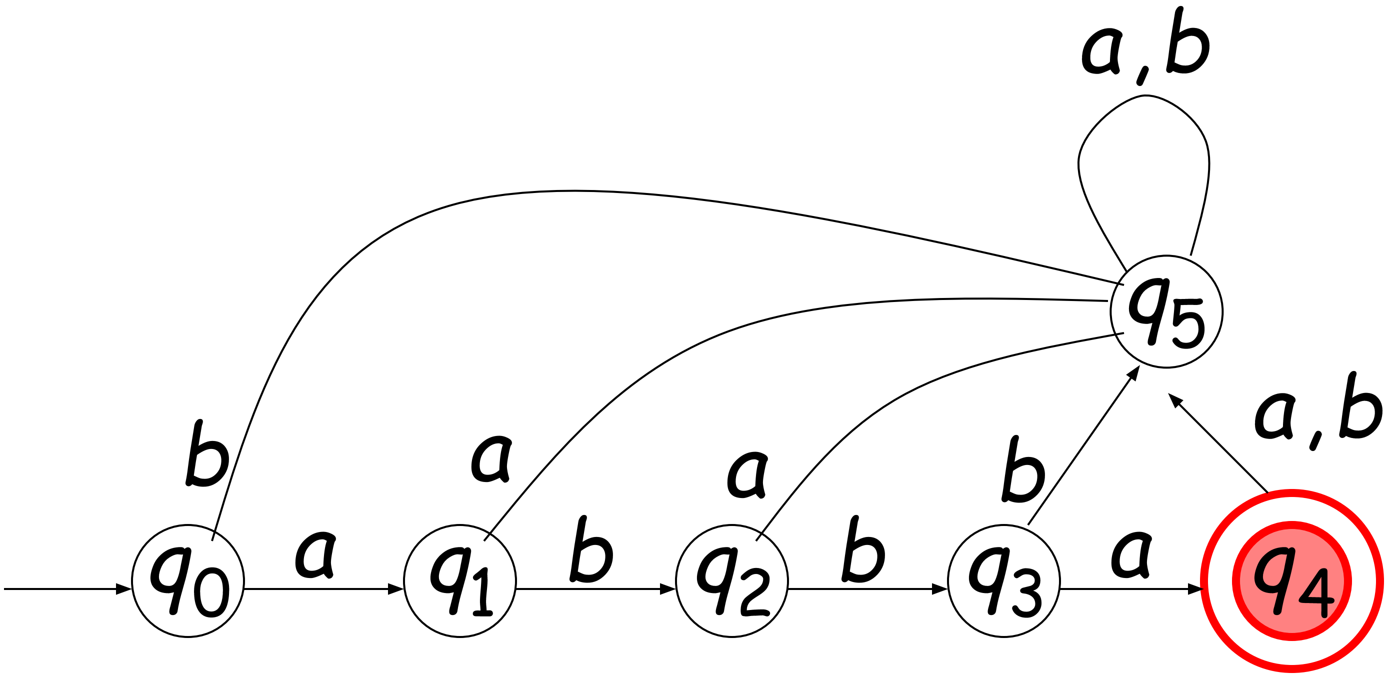
q_0



Set of Final States

F

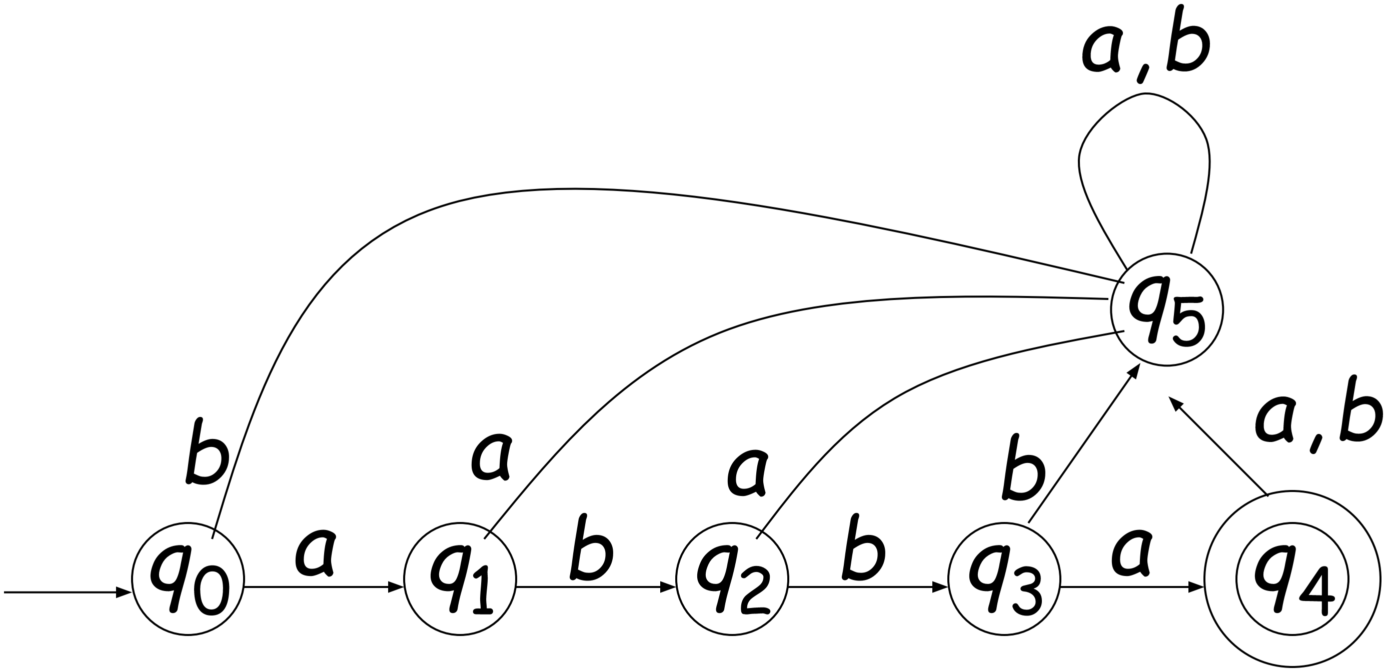
$$F = \{q_4\}$$



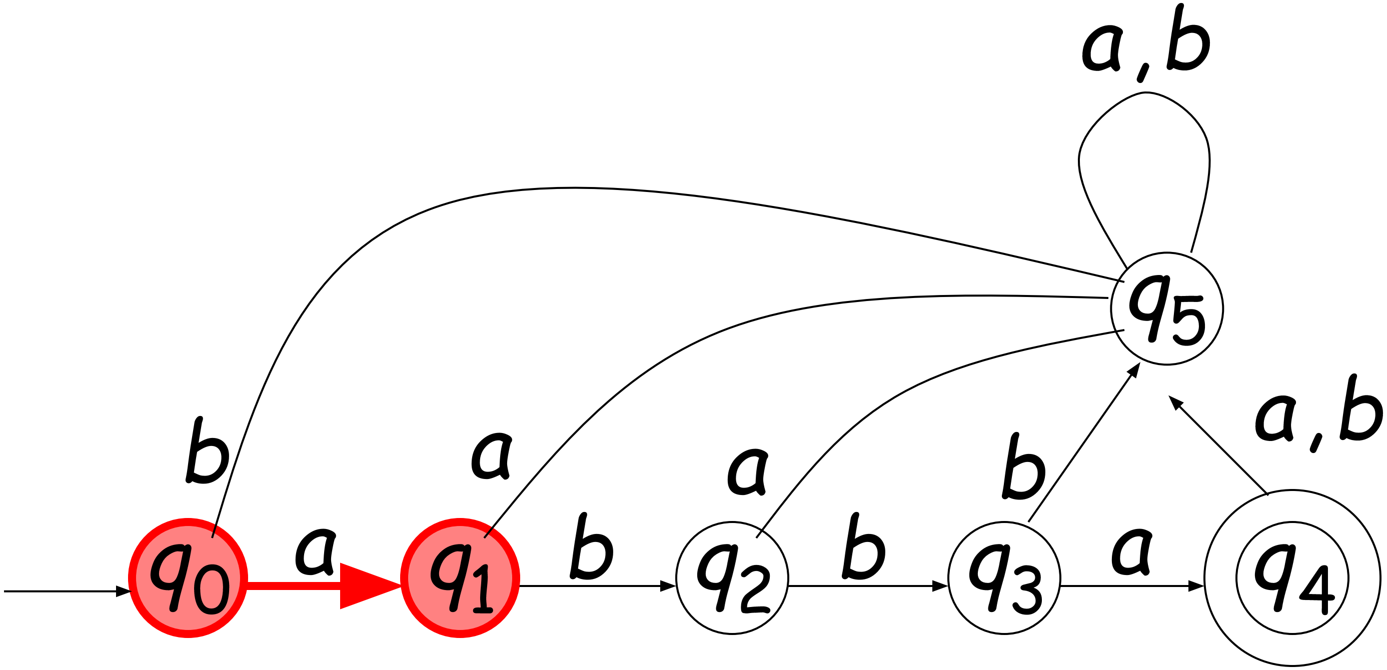
Transition Function

δ

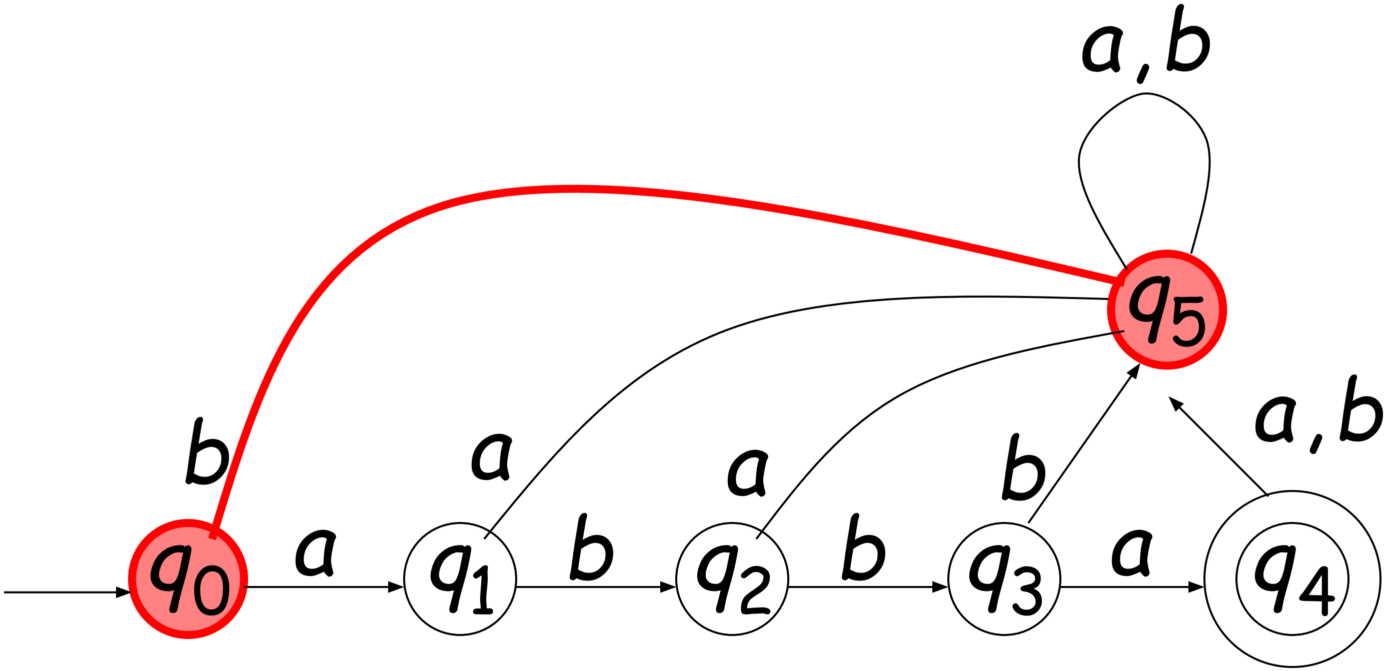
$$\delta : Q \times \Sigma \to Q$$



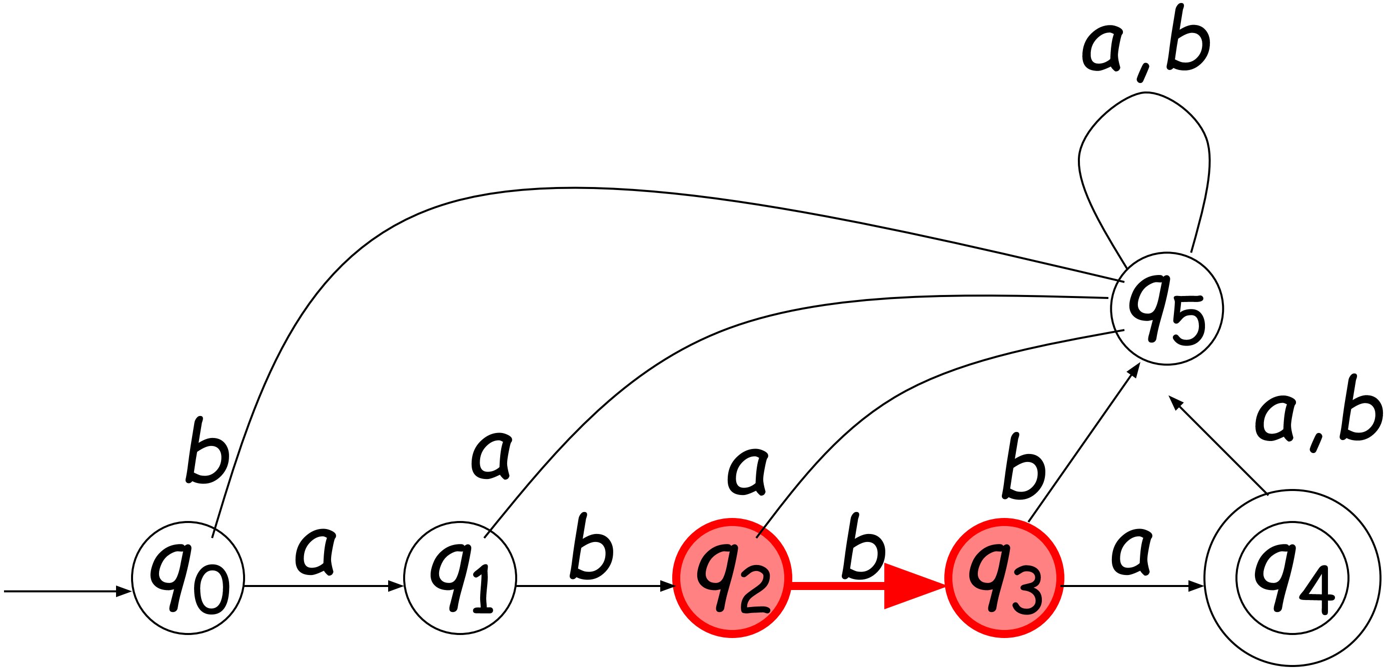
$$\delta(q_0, a) = q_1$$



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



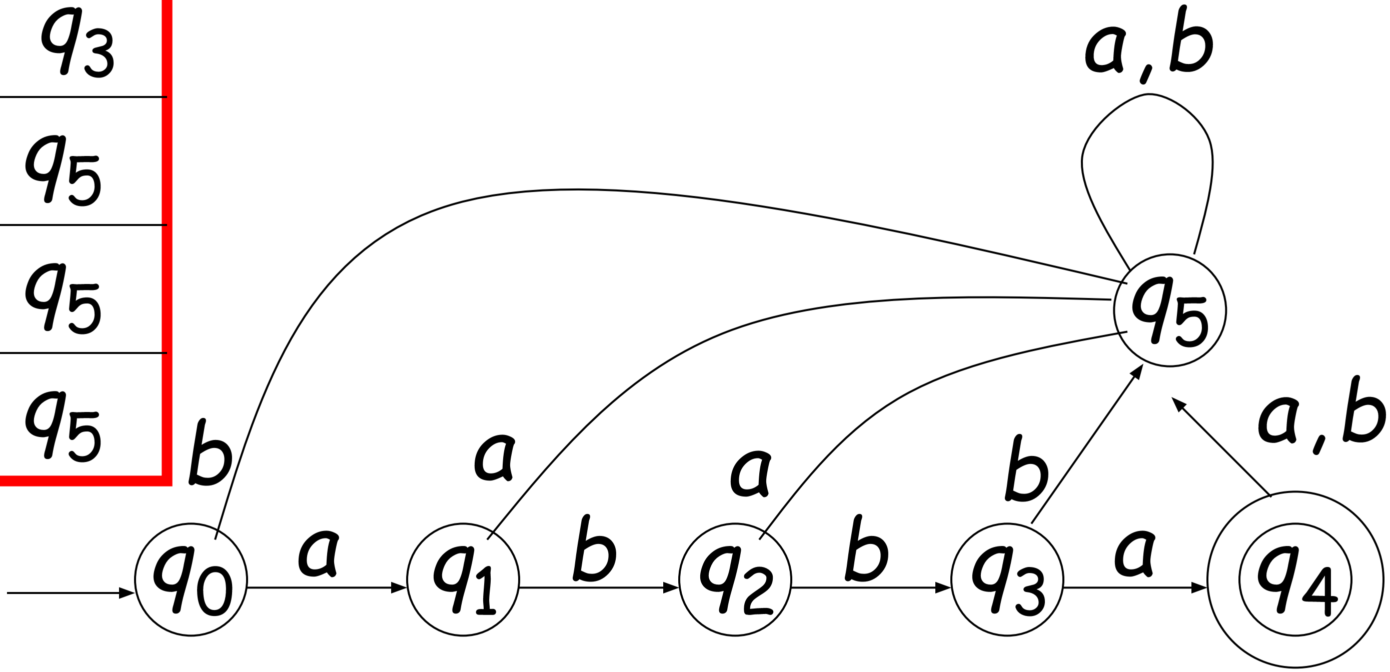
$$\delta(q_2, b) = q_3$$



Transition Function

δ

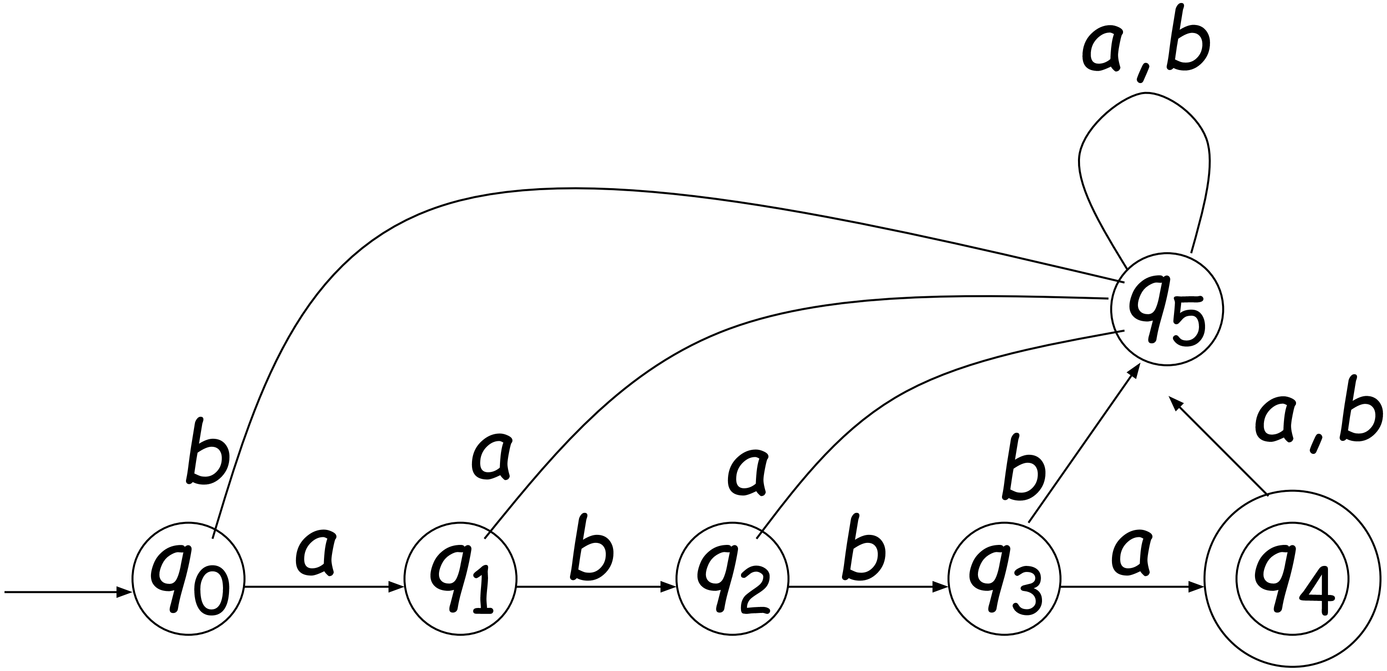
δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_2	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5



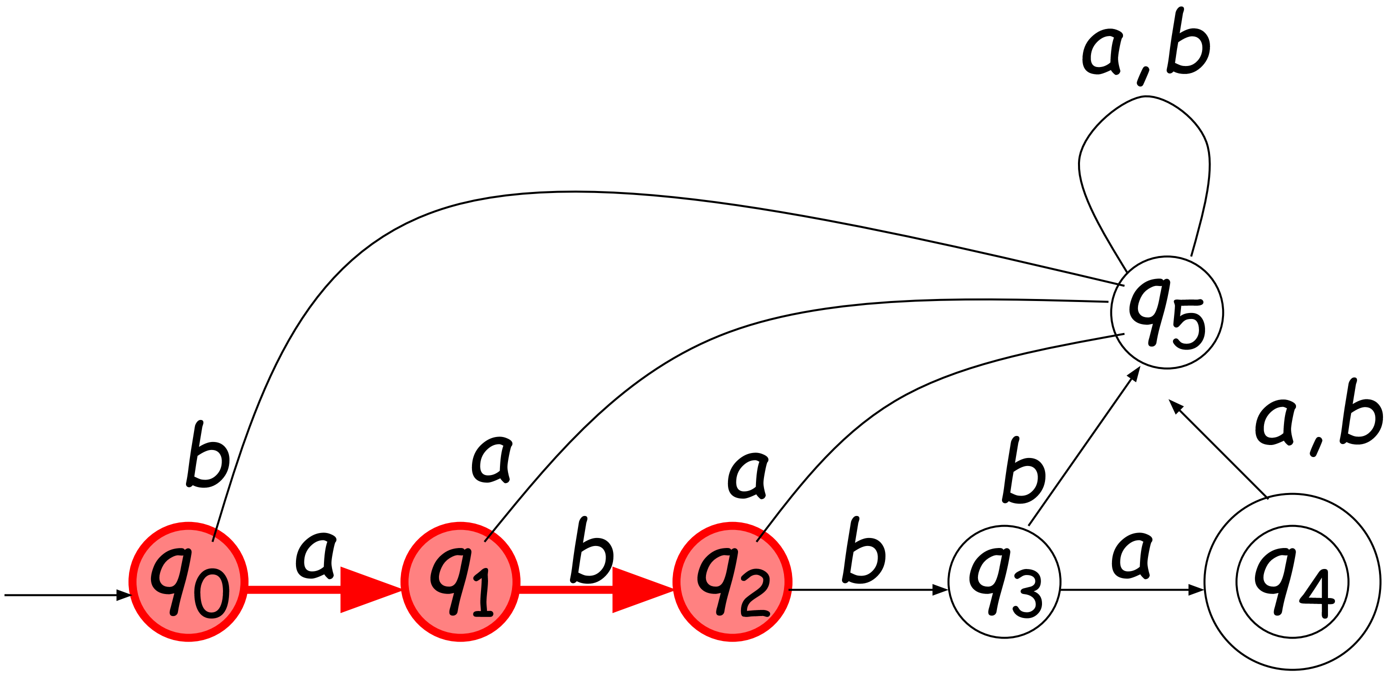
Extended Transition Function (Reads the entire string)

$\delta *$

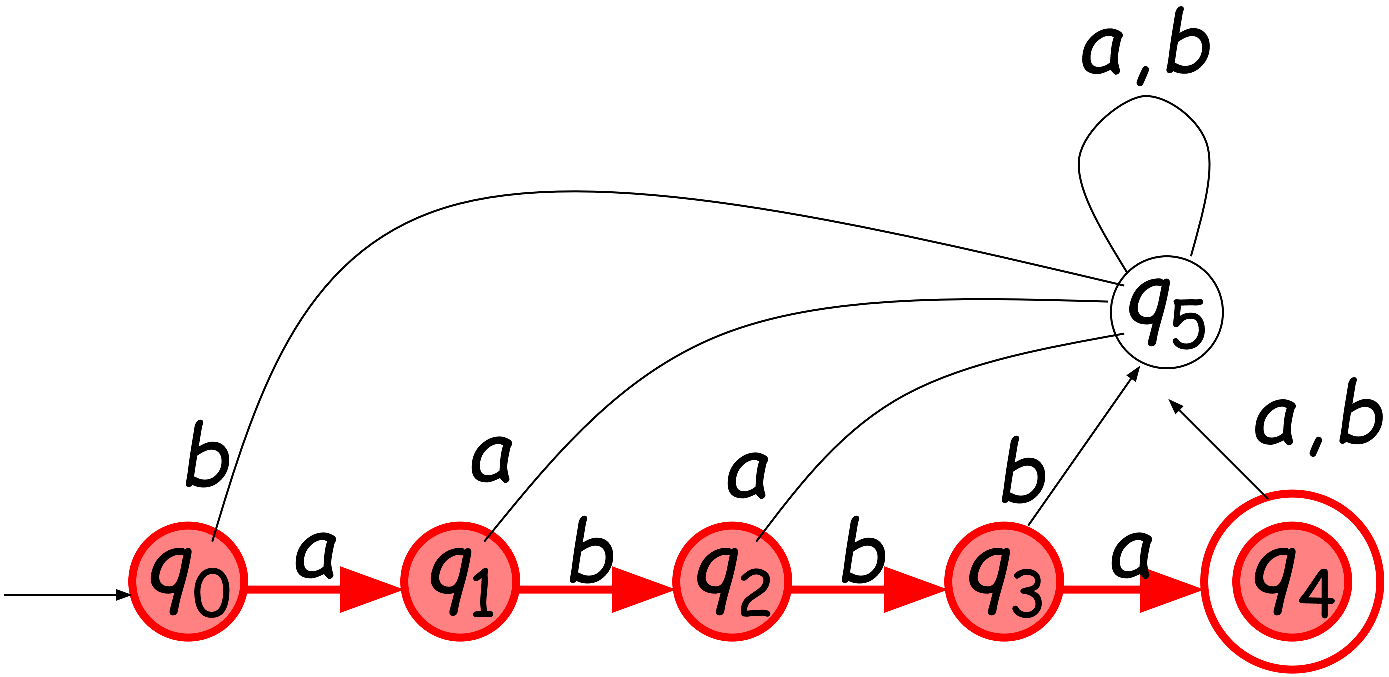
$$\delta * : Q \times \Sigma * \to Q$$



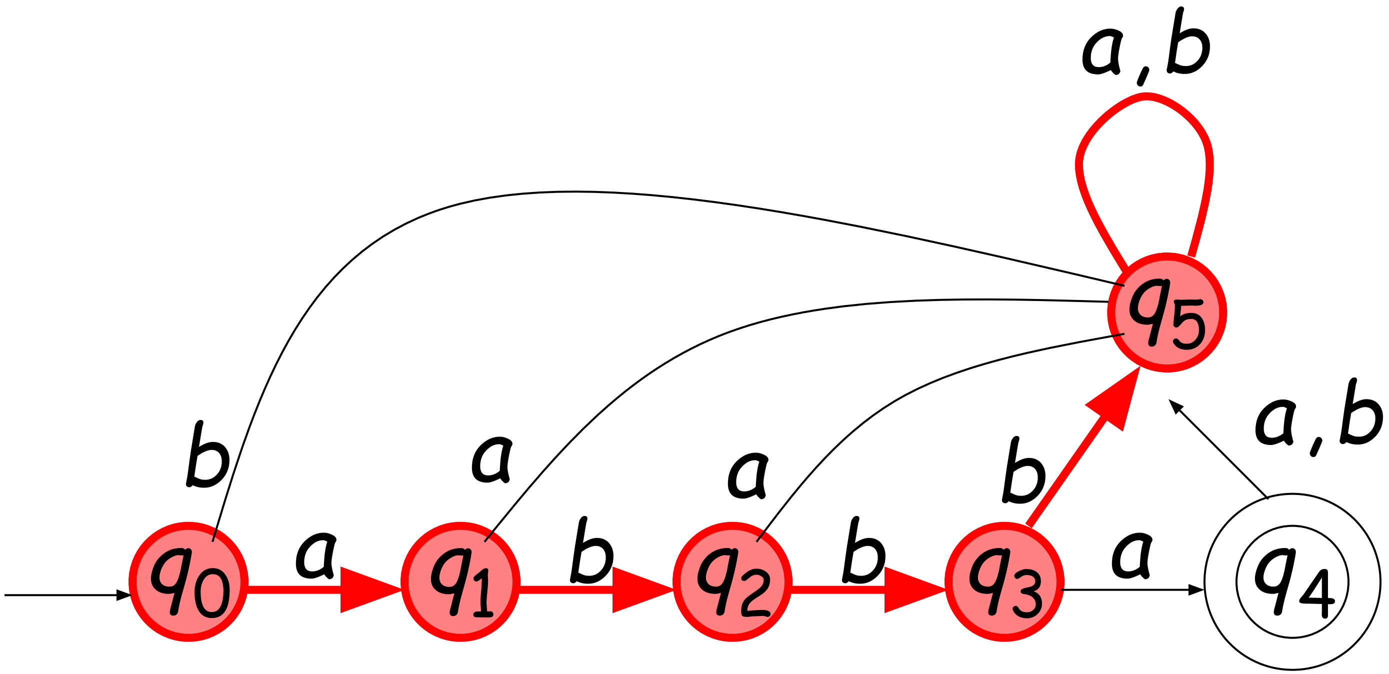
$$\delta^*(q_0, ab) = q_2$$



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

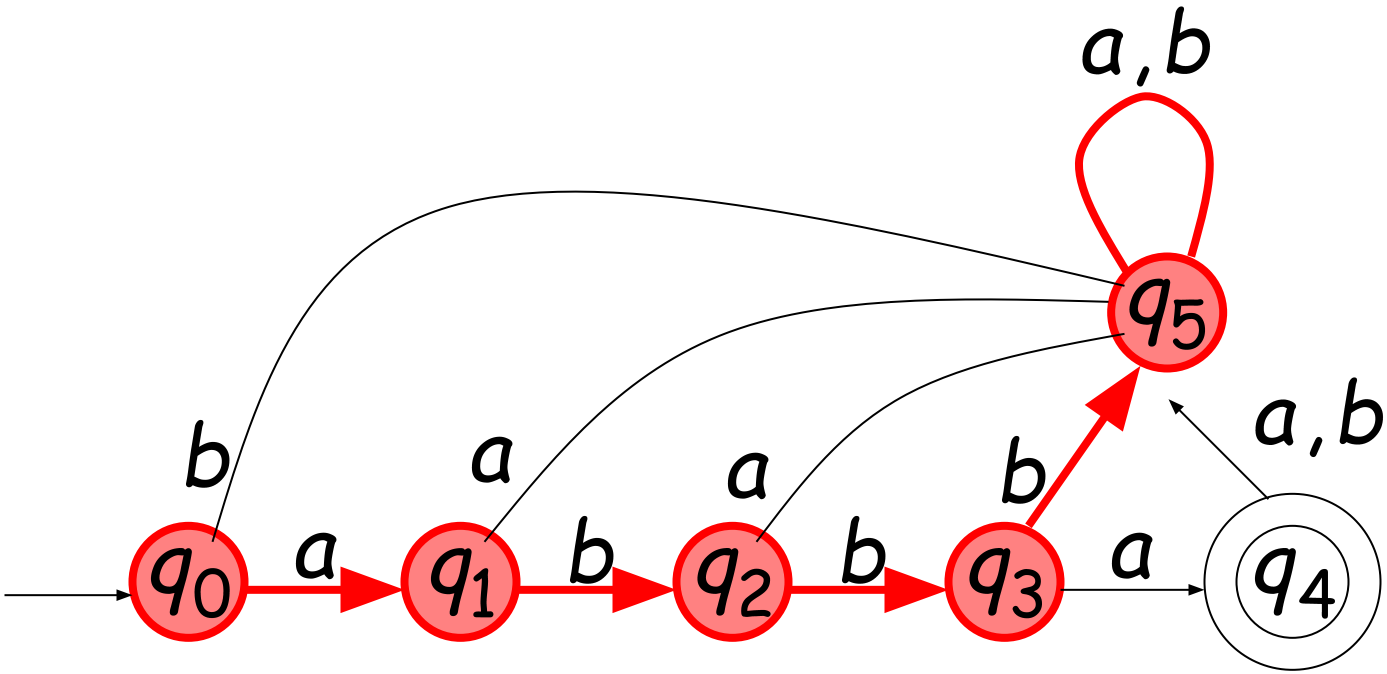


$$\delta^*(q_0, abbbaa) = q_5$$



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

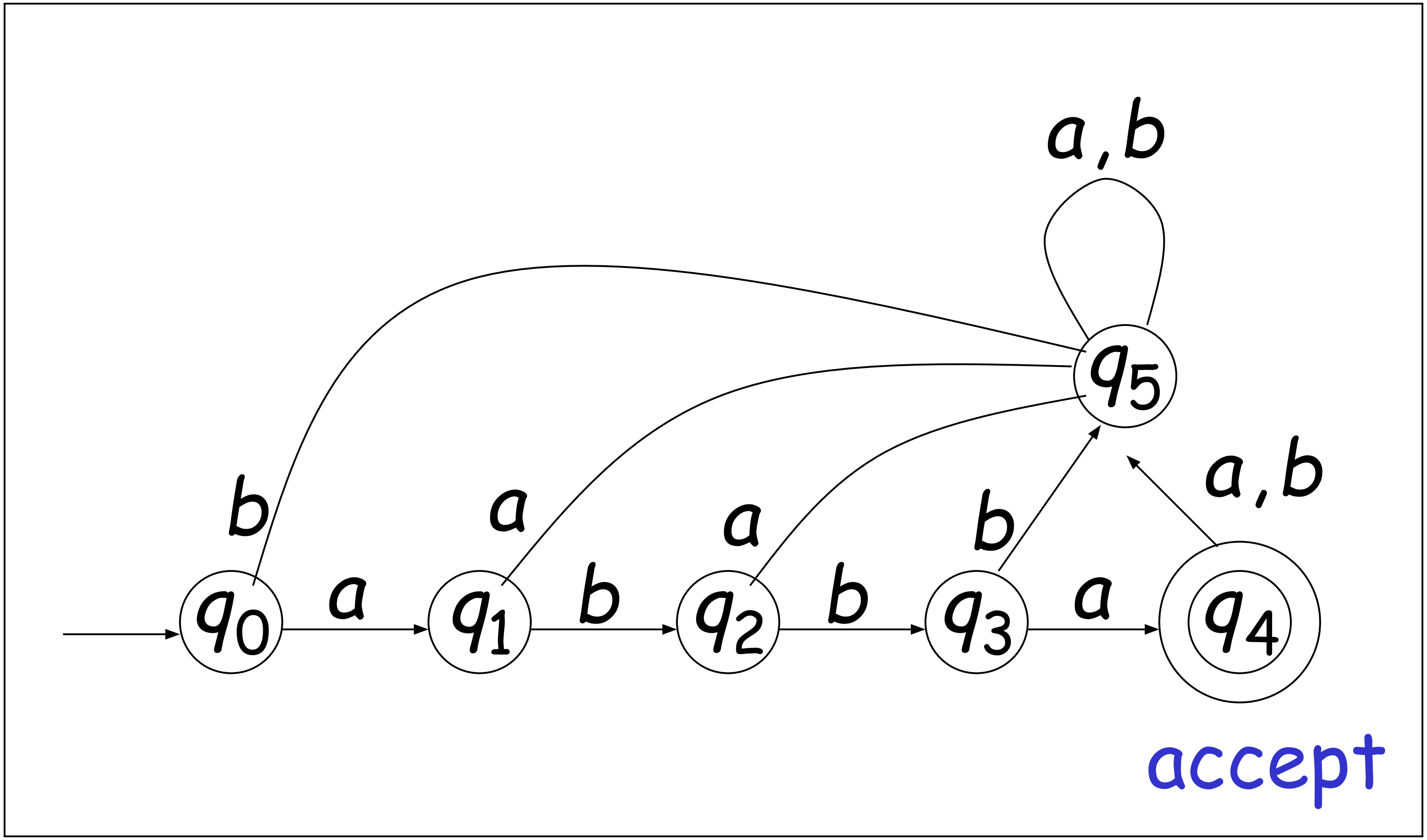
$$\delta^*(q_0, abbbaa) = q_5$$



Example

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

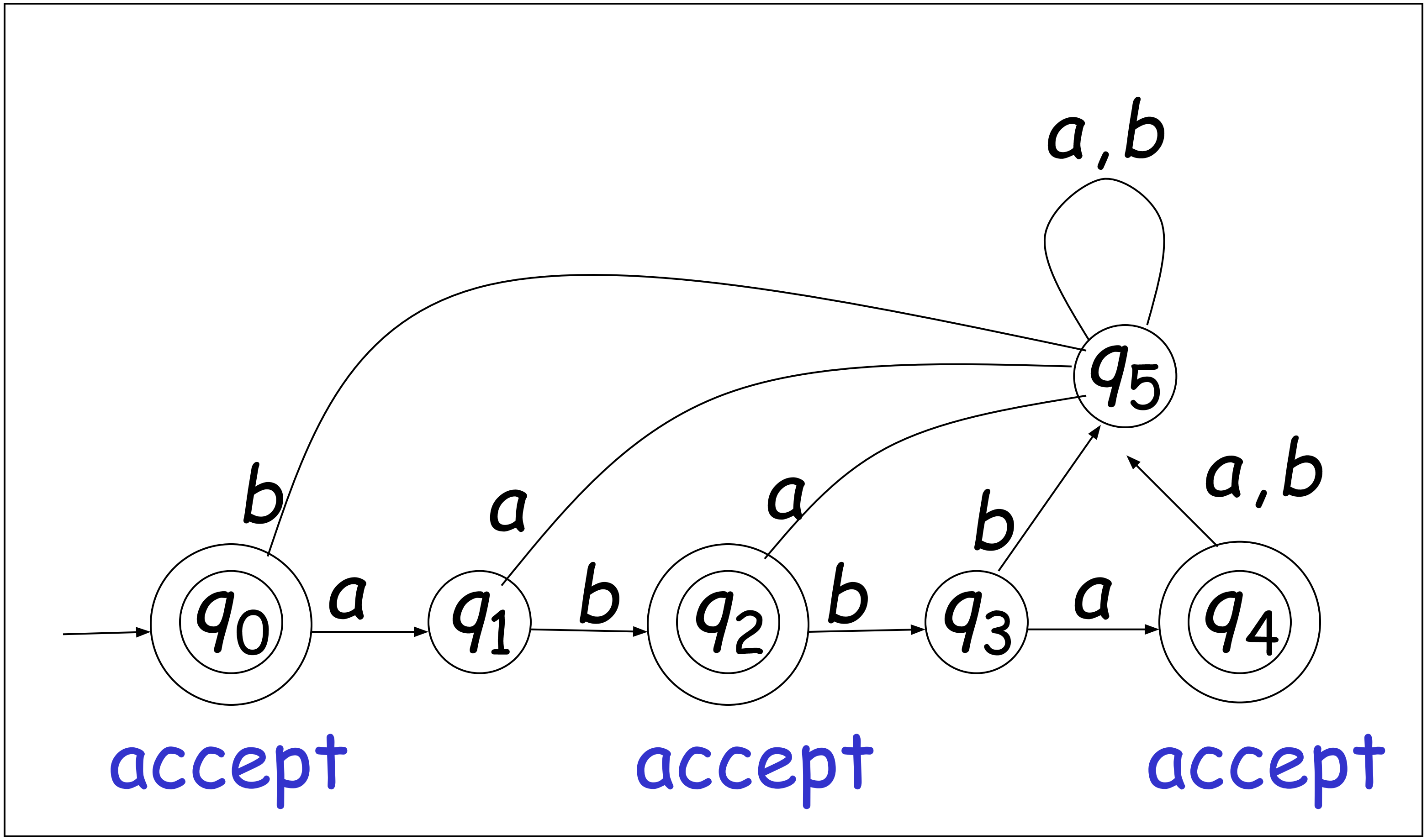
M



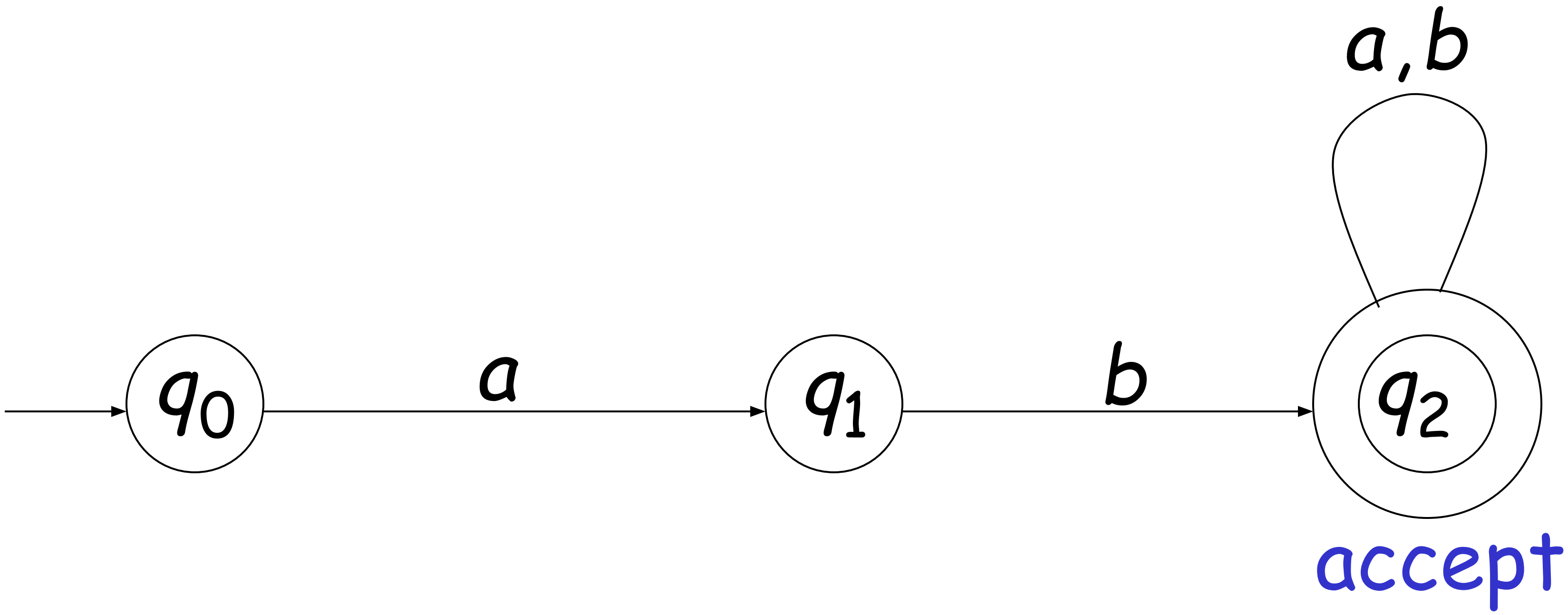
Another Example

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

M



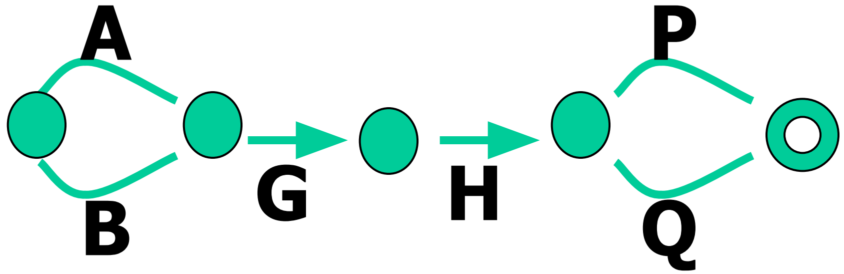
$L(M)$= { all substrings with prefix **ab** }



FSA & RE

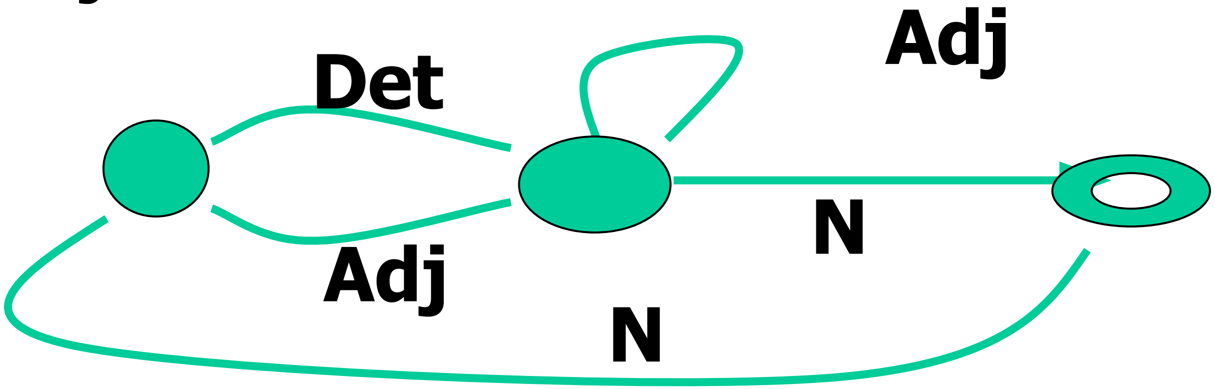
X → (A | B) G H (P | Q)

X →



NP → (Det | ε) Adj* N

NP →

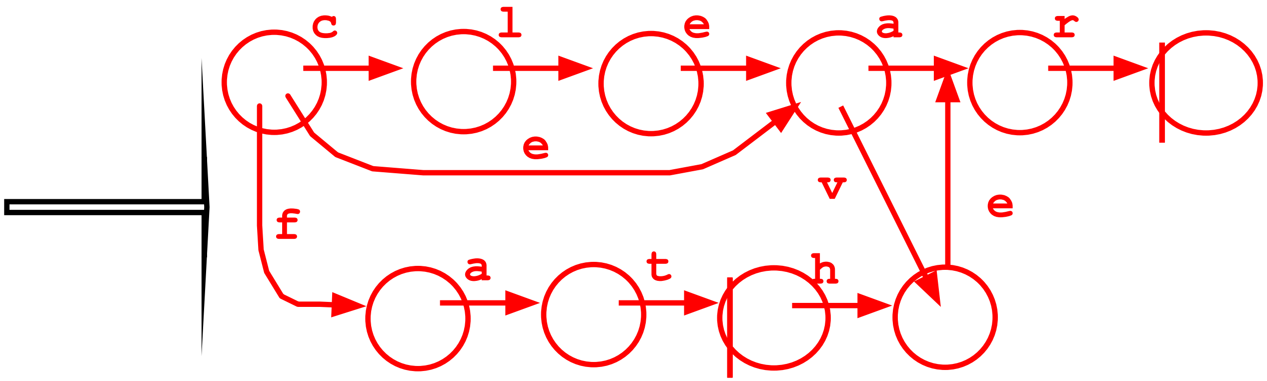


FSA & RE

Wordlist

clear
clever
ear
ever
fat
father

Network



Ref: L. Karttunen