CSC 339 – Theory of Computation
Fall 2022-2023

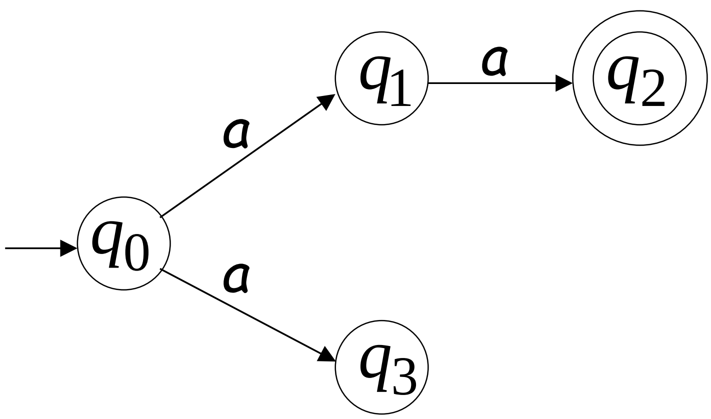
4. Nondeterministic Finite Automata

Outline

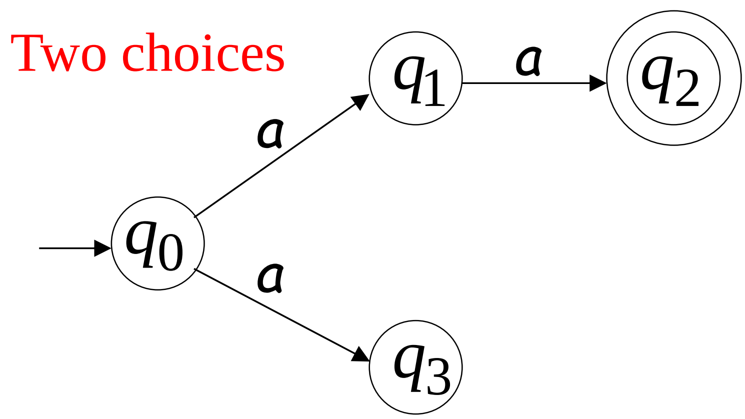
- Nondeterministic Finite Automata (NFA)
- Examples
- Epsilon transition
- Formal definition
- Languages accepted by NFA and DFA
- Conversion of NFA to DFA
- NFA and DFA are equivalent

Nondeterministic Finite Automata (NFA)

Alphabet = $\{a\}$

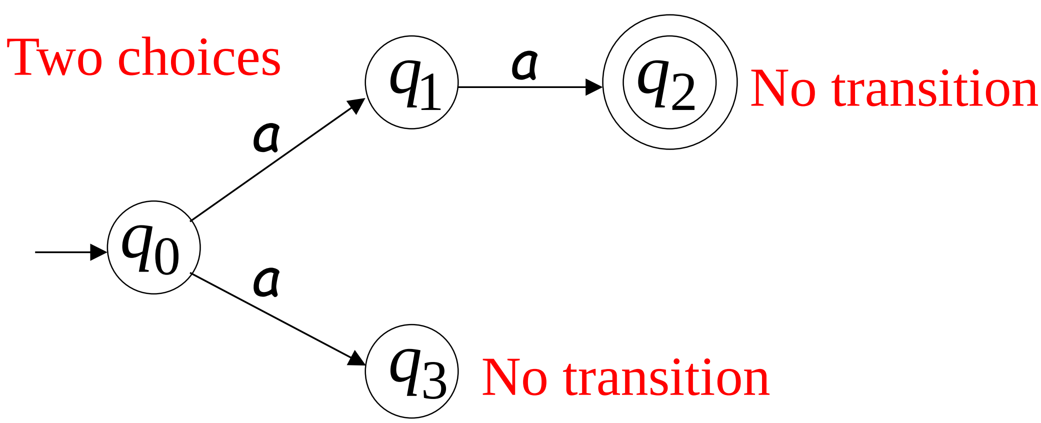


Alphabet $= \{a\}$

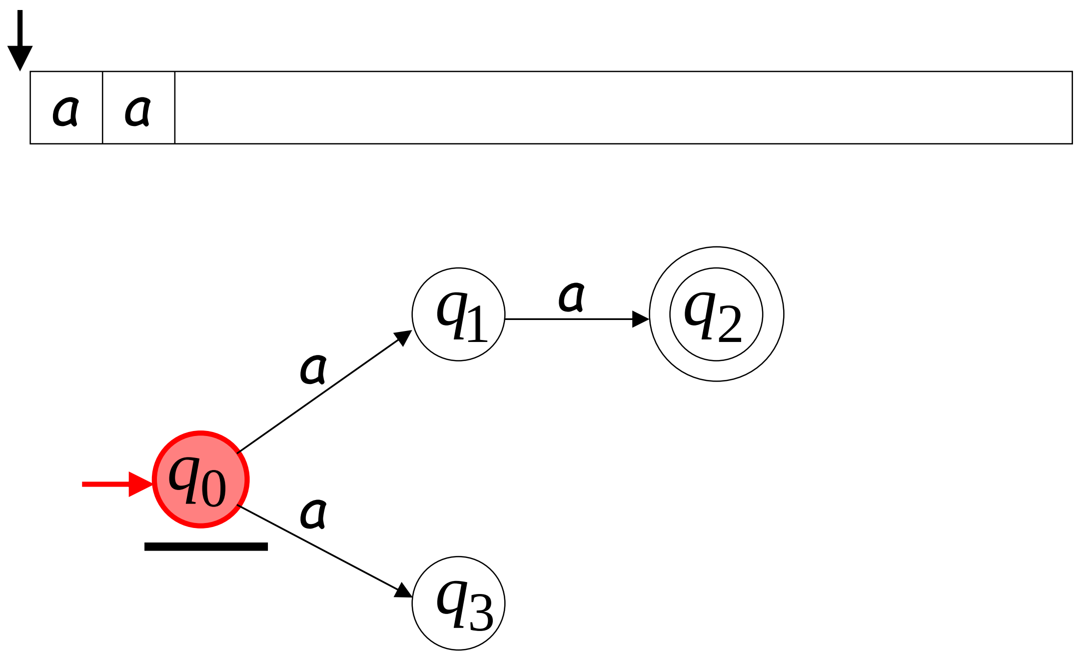


4

Alphabet = $\{a\}$

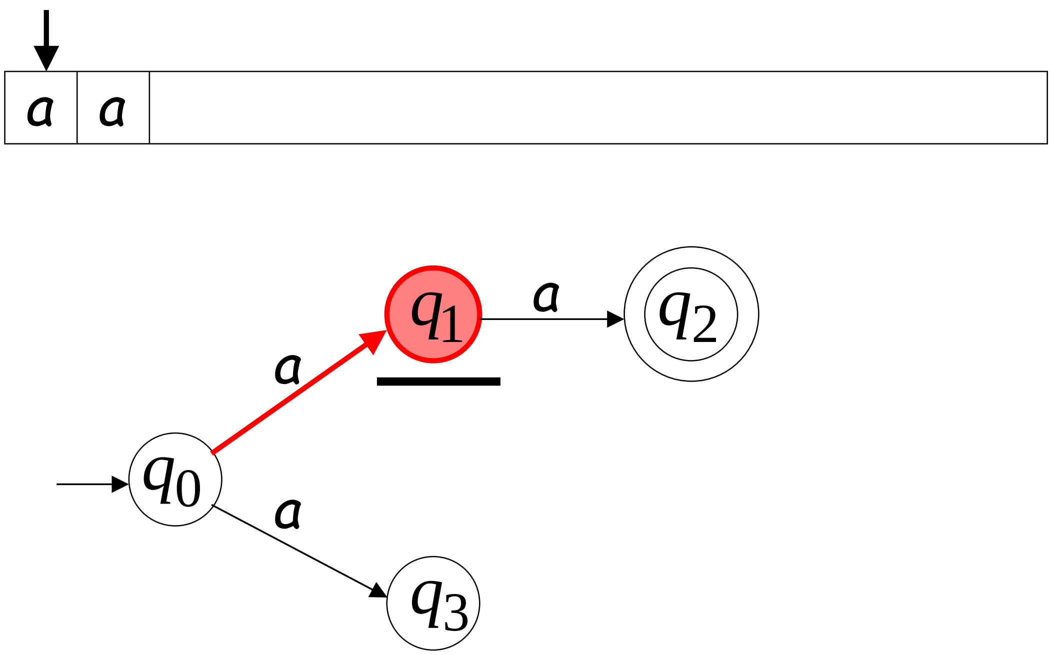


First Choice



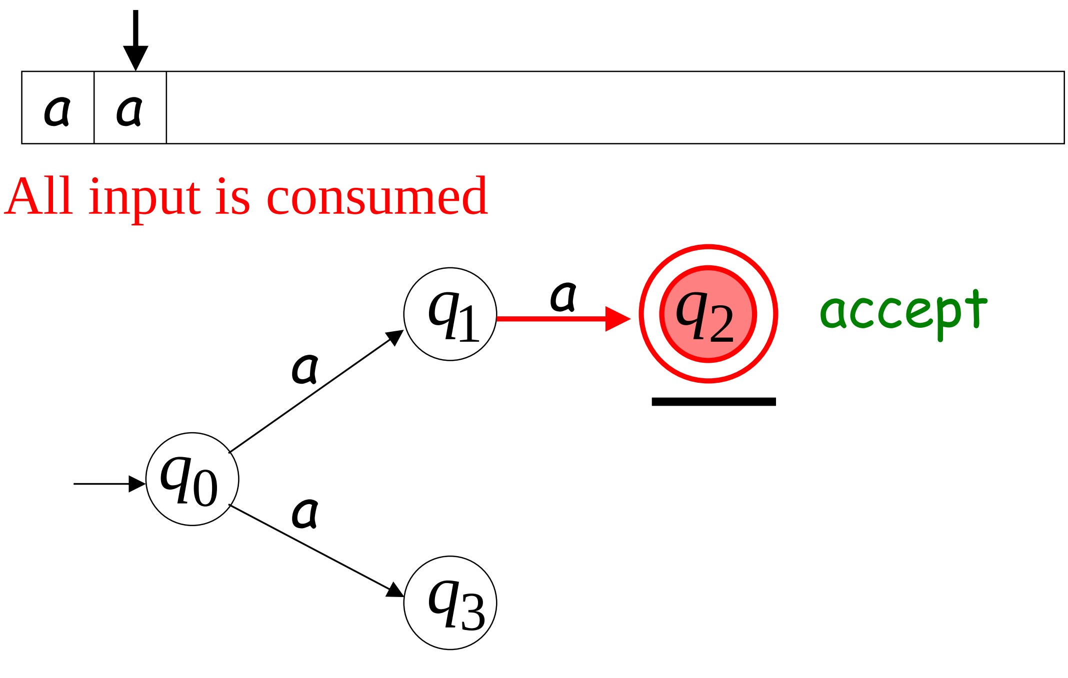
6

First Choice



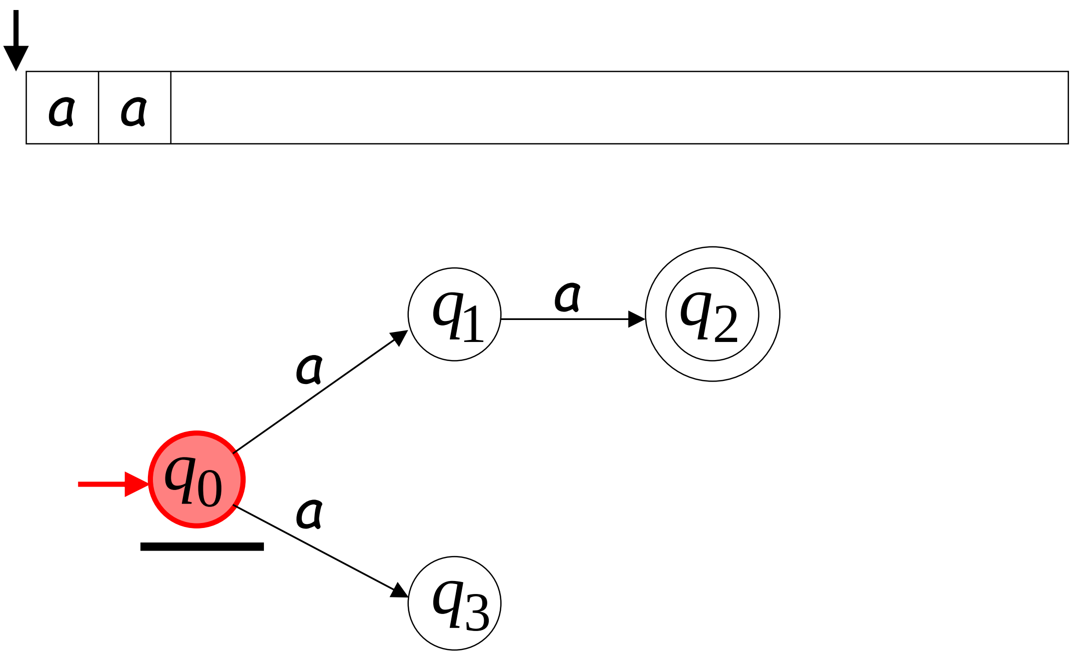
7

First Choice



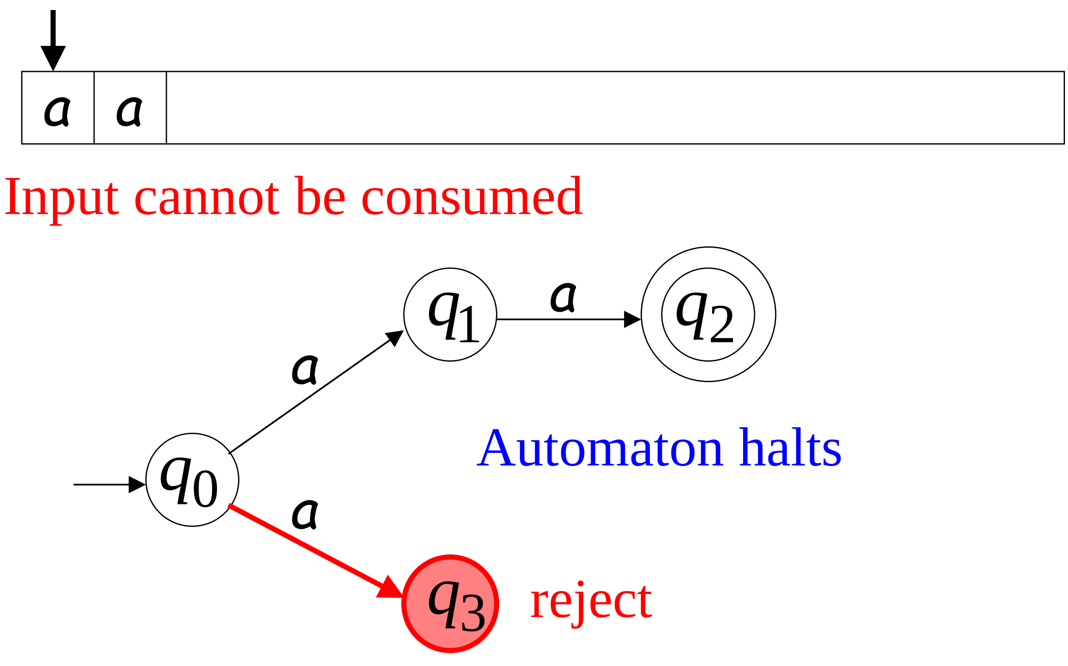
8

Second Choice



9

Second Choice



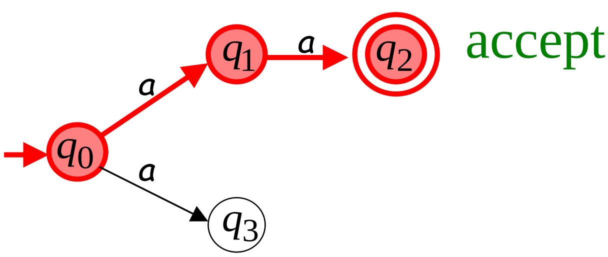
10

Nondeterministic Finite Automata (NFA)

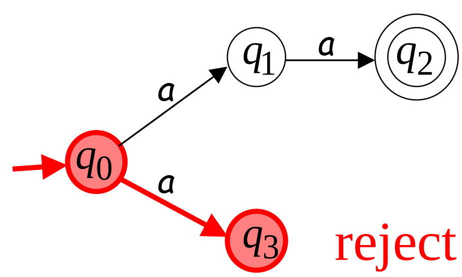
An NFA accepts a string:

If there is a computation of the NFA that accepts the string i.e., all the input string is processed and the automaton is in an accepting state.

aa is accepted by the NFA:



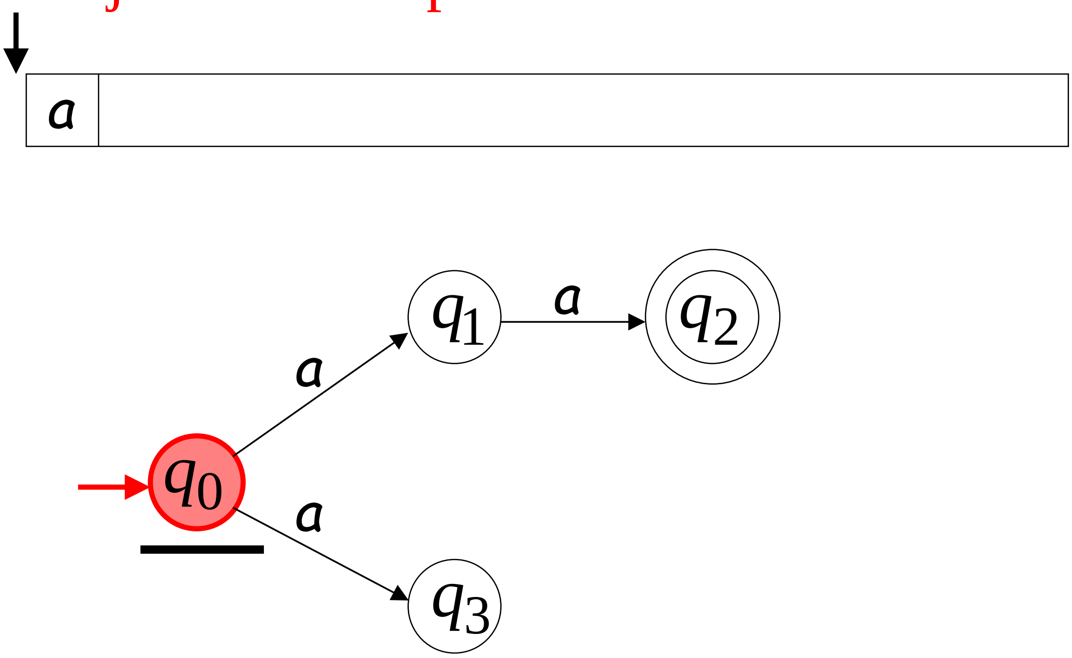
because this computation accepts *aa*



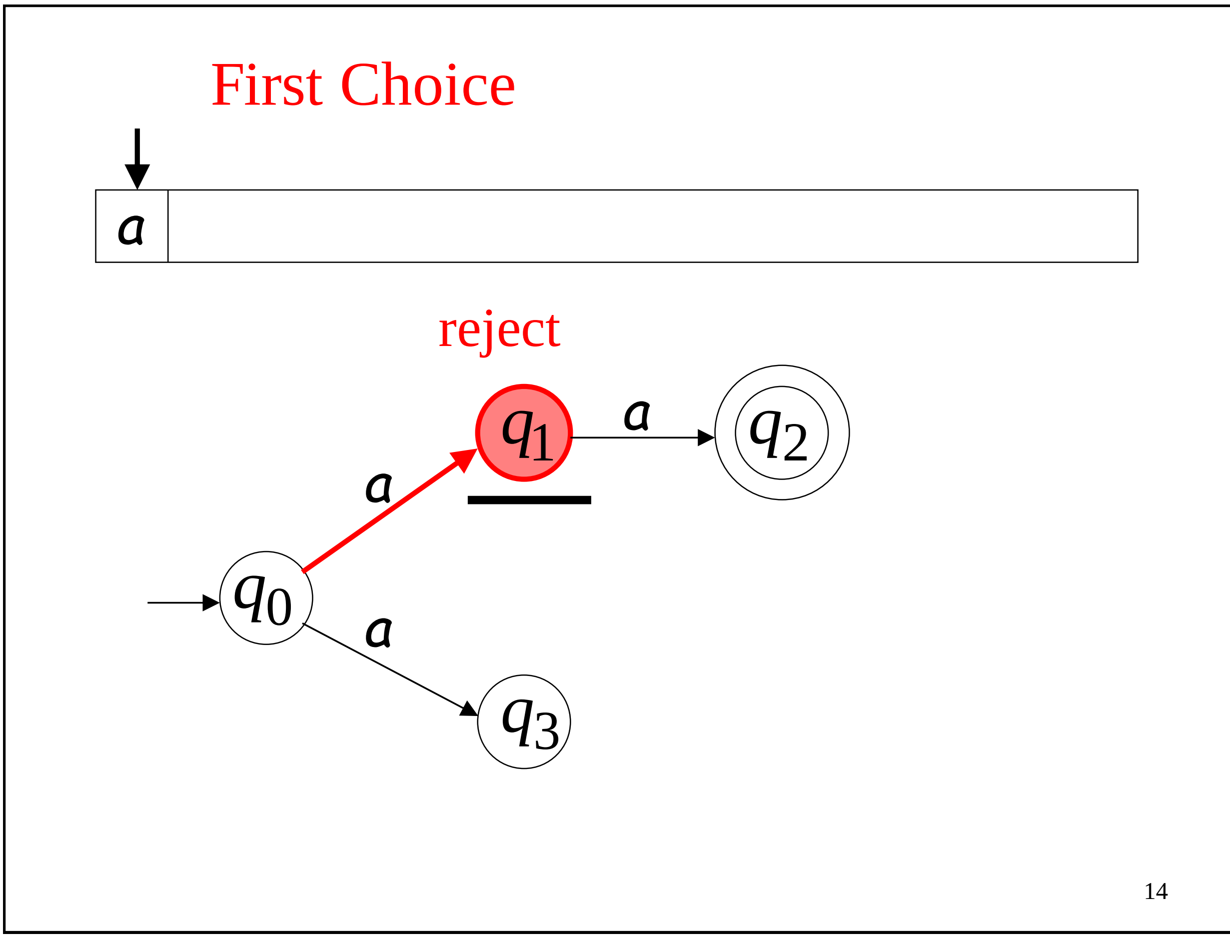
this computation is ignored

12

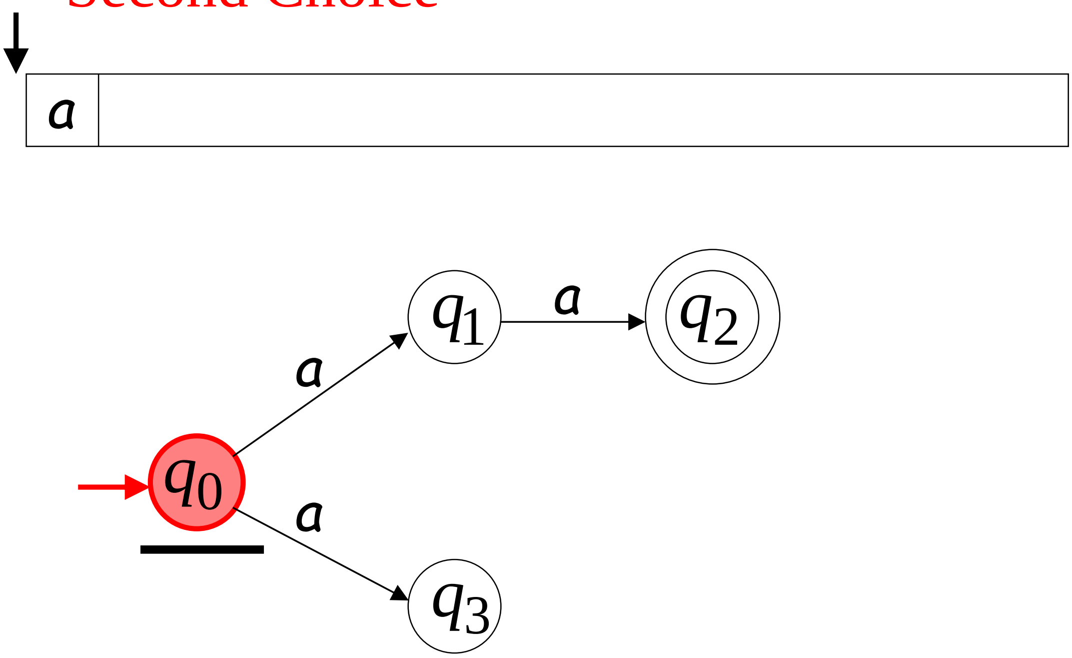
Rejection example



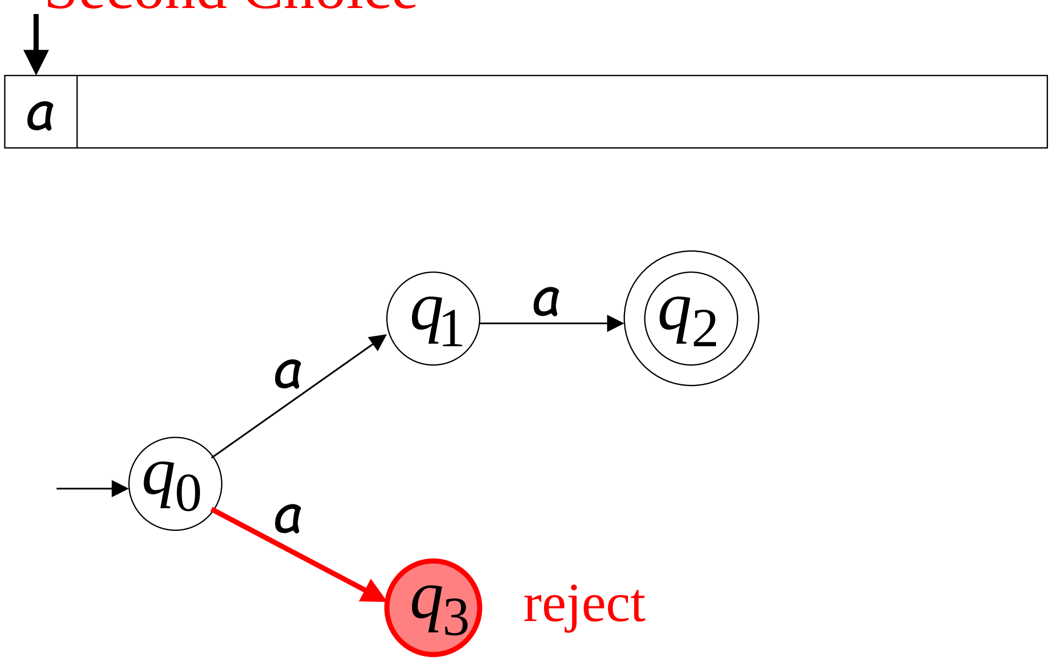
13


First Choice
a
reject
a
a
a
q_0
q_1
q_2
q_3
14

Second Choice



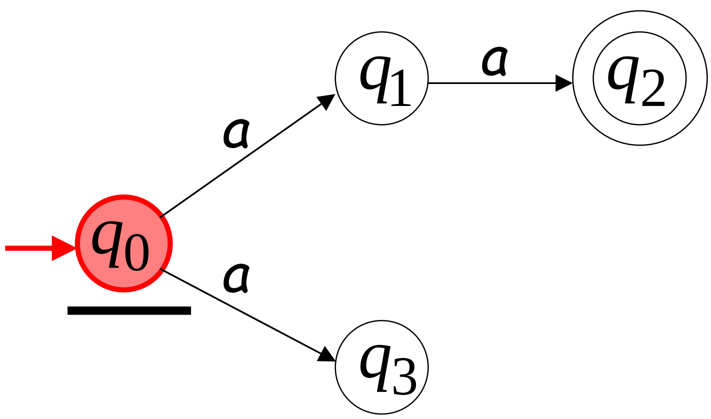
Second Choice

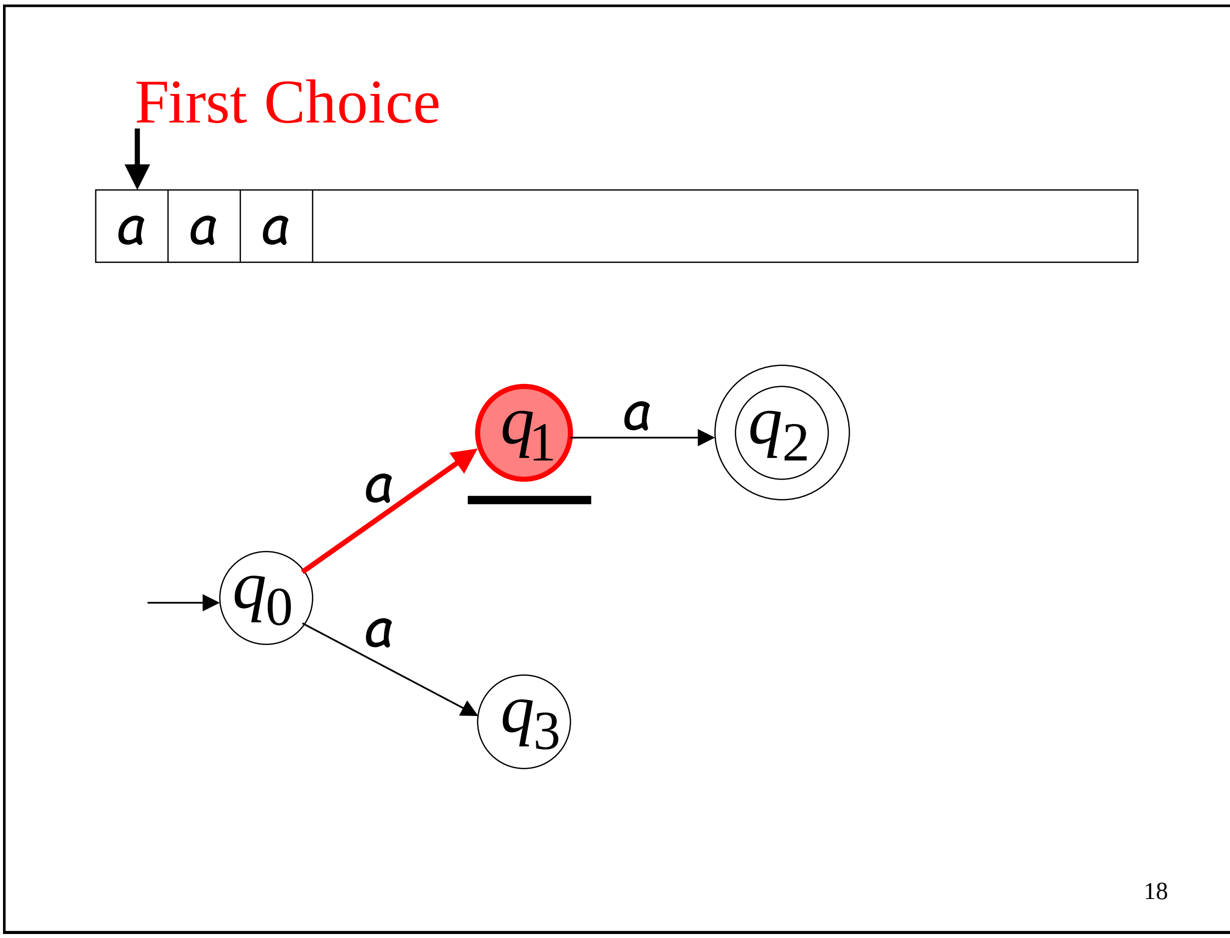


16

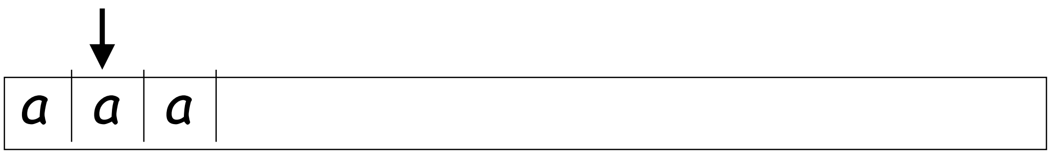
Another rejection example

a	a	a	

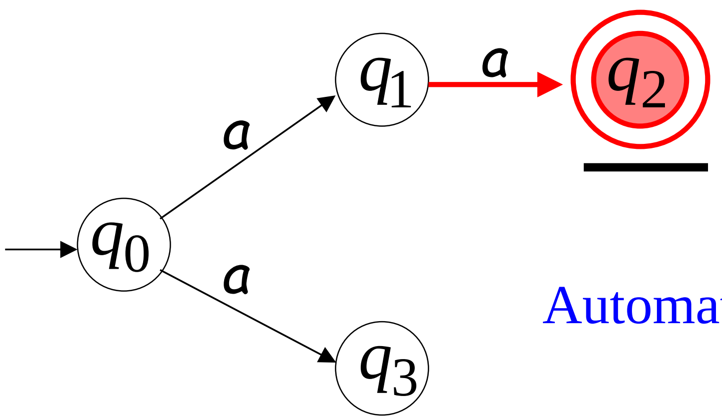



First Choice
a a a
a
a
a
q_0
q_1
q_2
q_3
18

First Choice



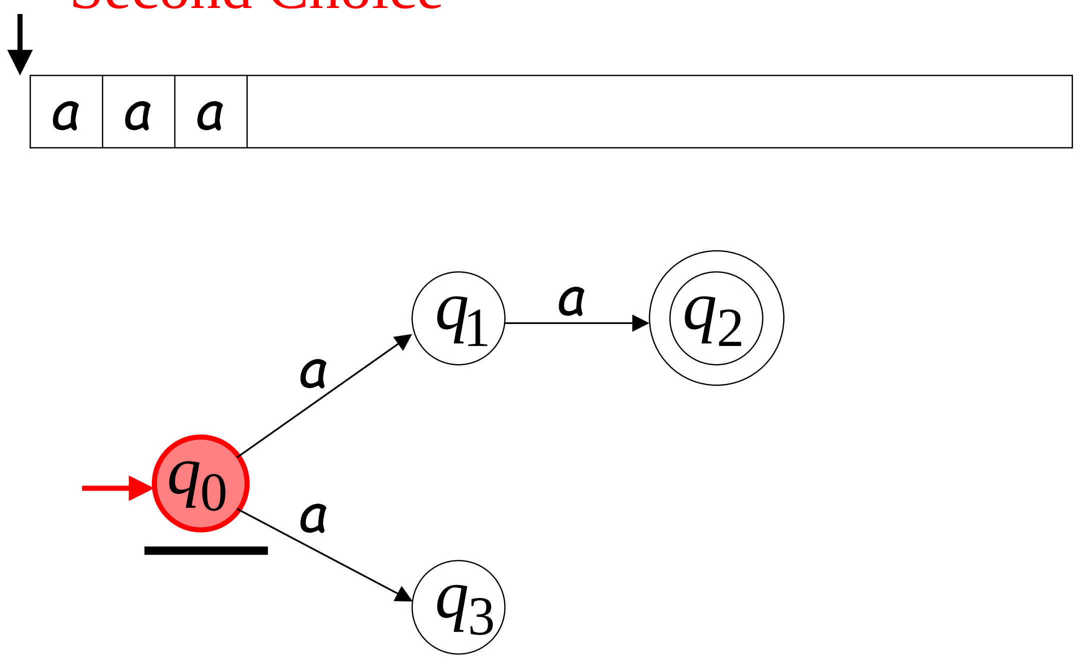
Input cannot be consumed



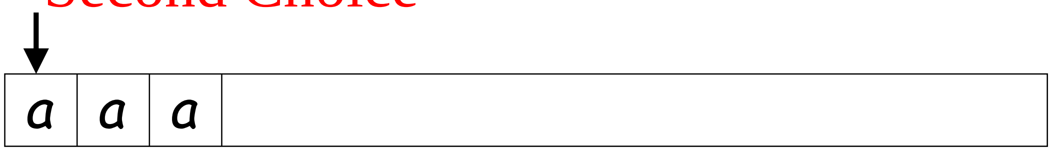
reject

Automaton halts

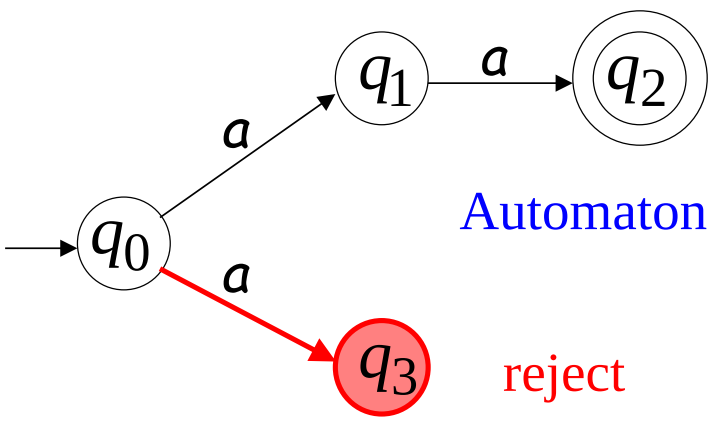
Second Choice



Second Choice



Input cannot be consumed



Automaton halts

reject

21

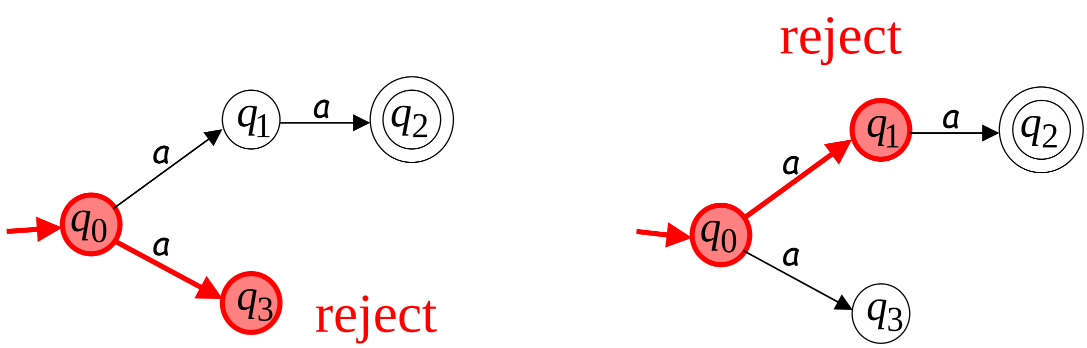
Nondeterministic Finite Automata (NFA)

An NFA rejects a string:

If there is no computation of the NFA that accepts the string i.e. for each computation:

- All the input is consumed and the automaton is in a non-final state, OR
- The input cannot be consumed.

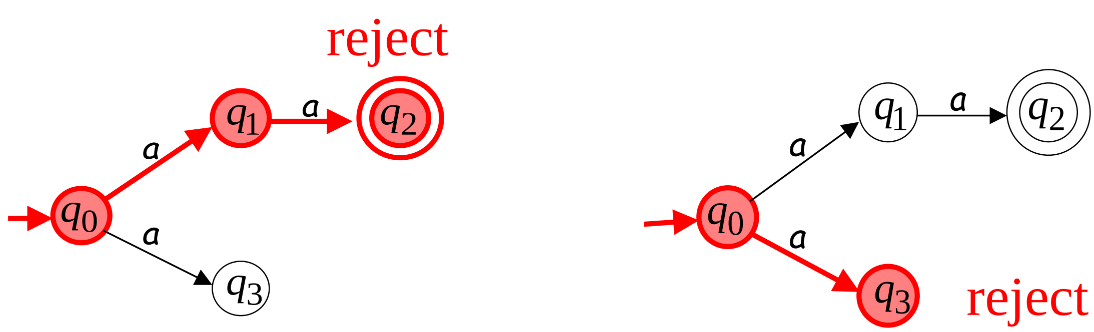
a is rejected by the NFA:



All possible computations lead to rejection

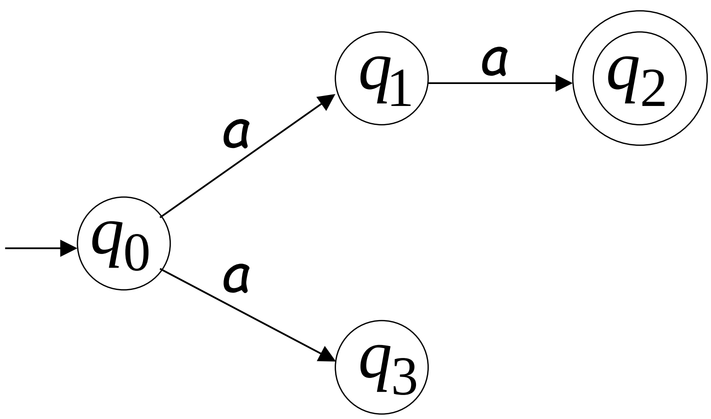
23

aaa is rejected by the NFA:



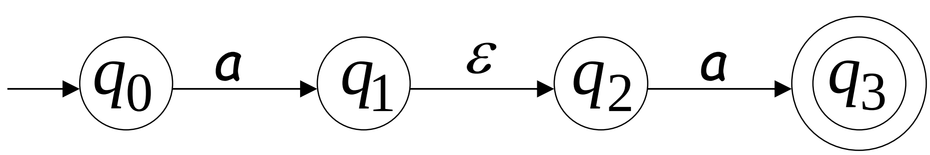
All possible computations lead to rejection

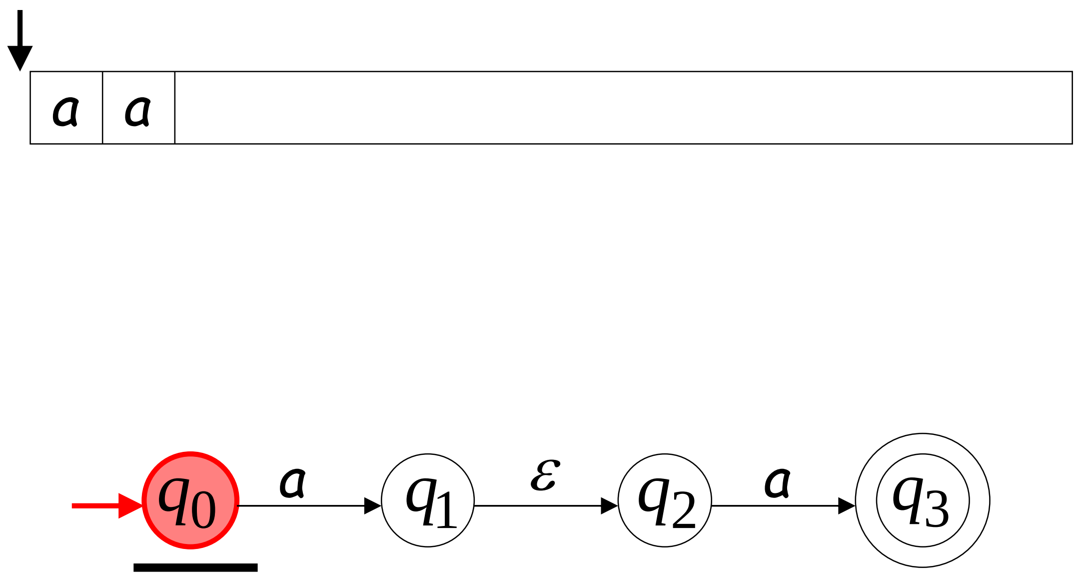
Language accepted: $L = \{aa\}$



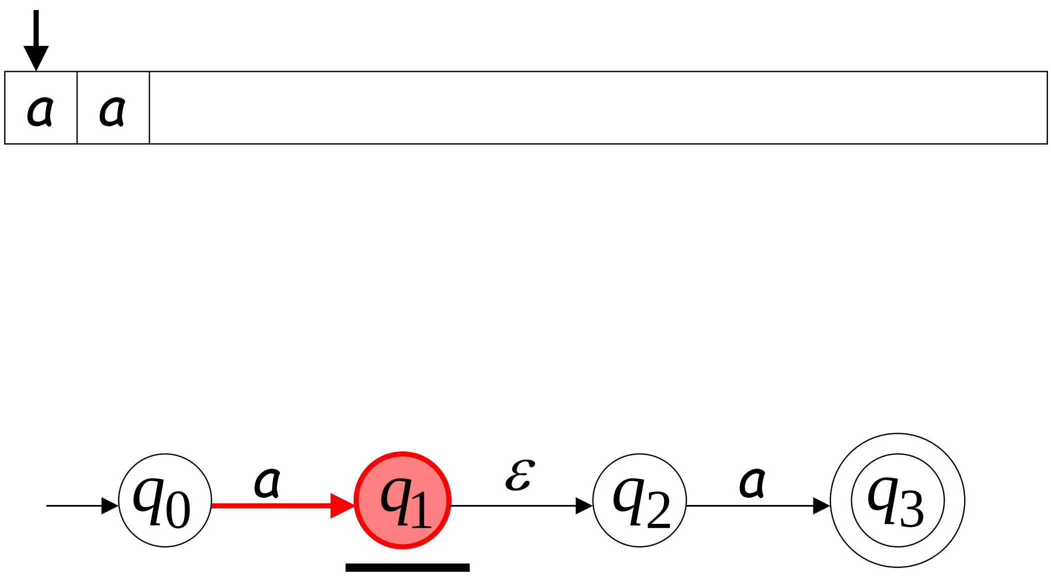
25

Lambda Transition

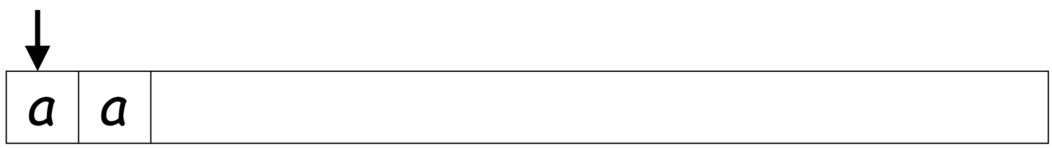




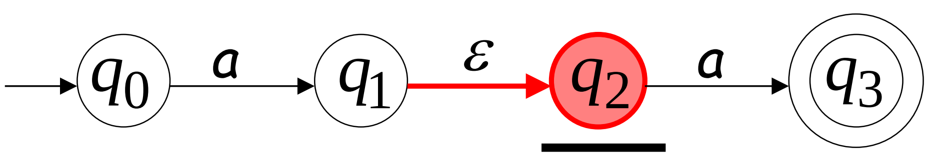
27

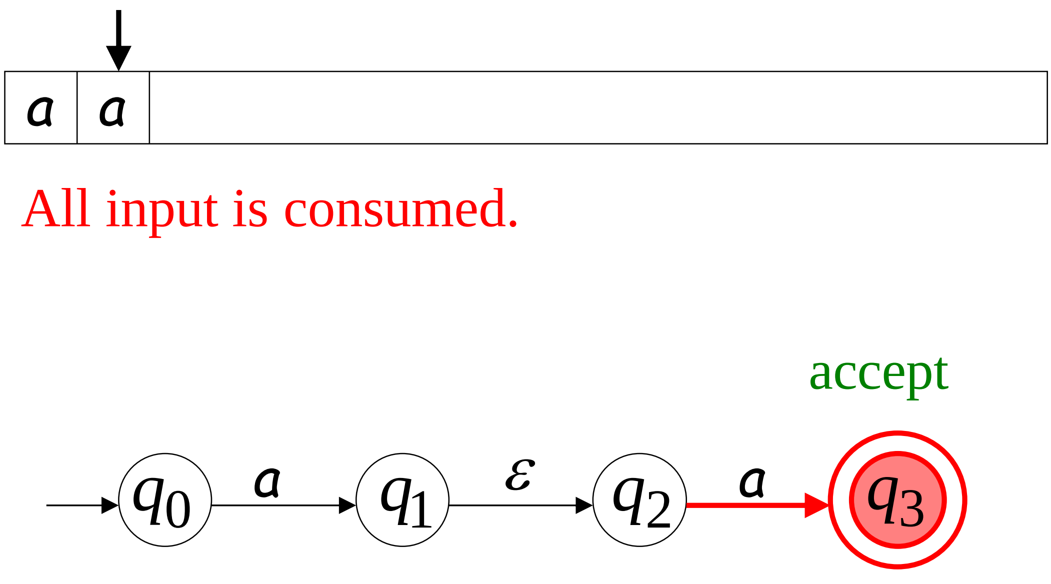


28



Input tape head does not move.

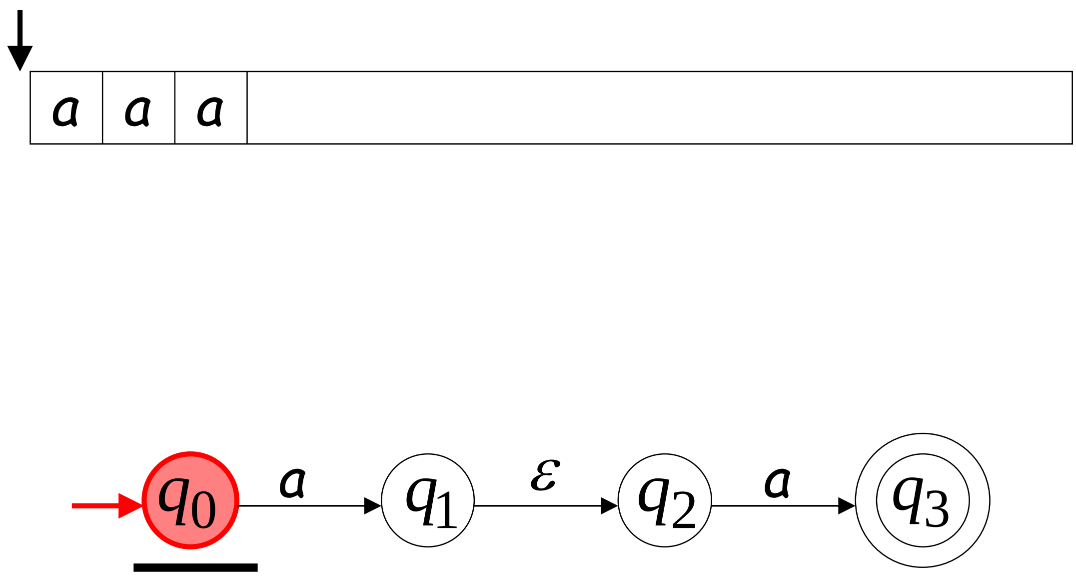


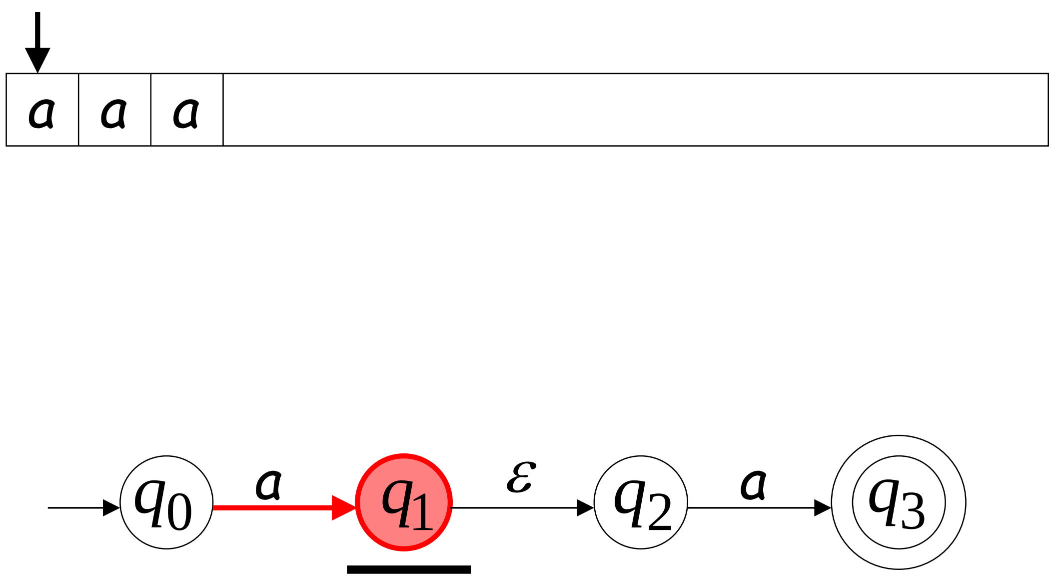


String aa is accepted

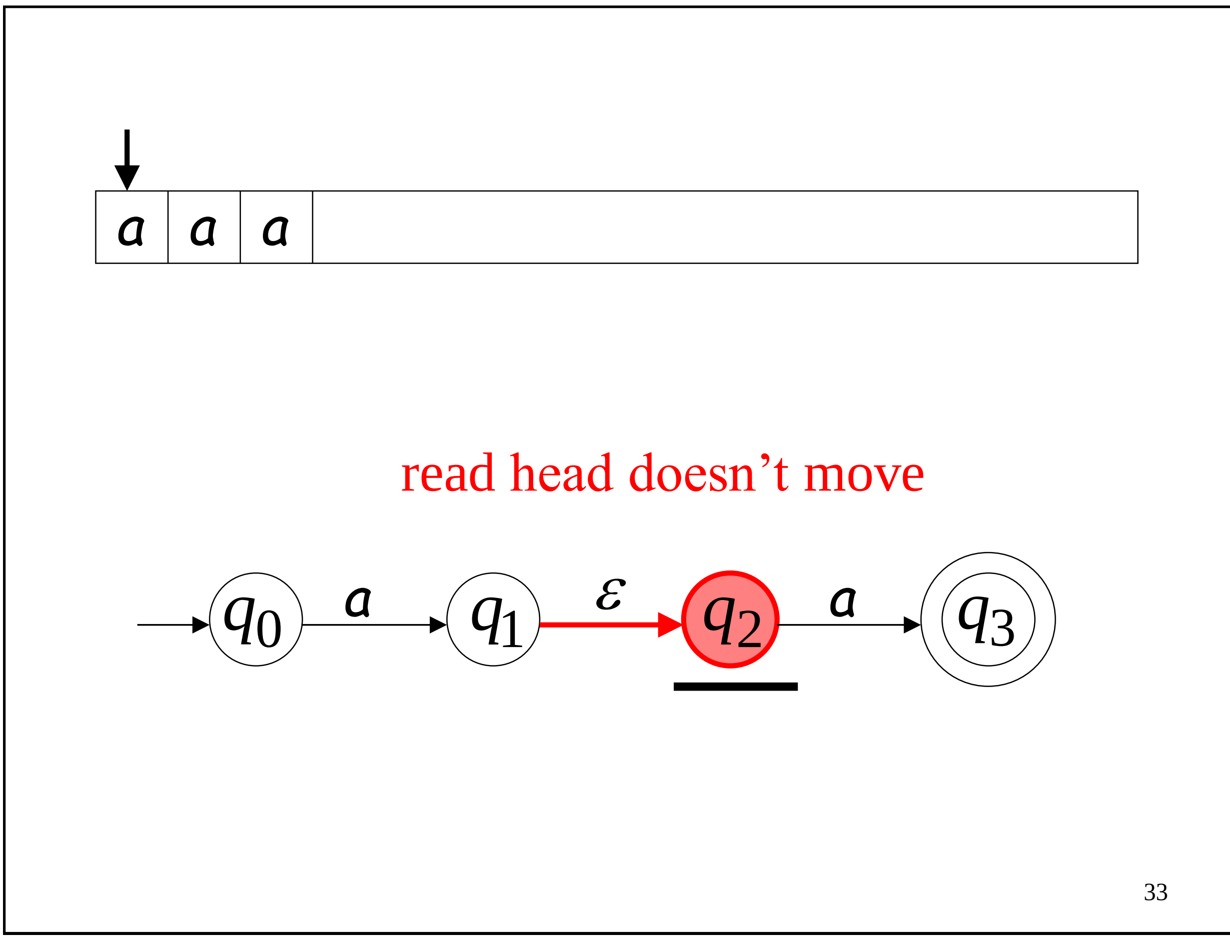
30

Rejection Example

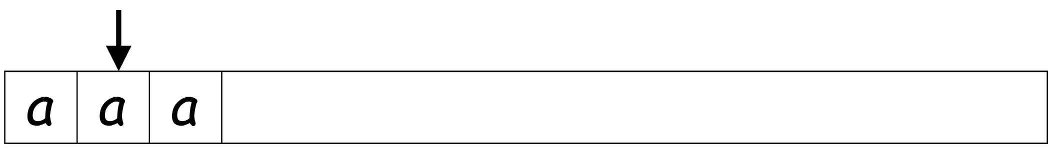




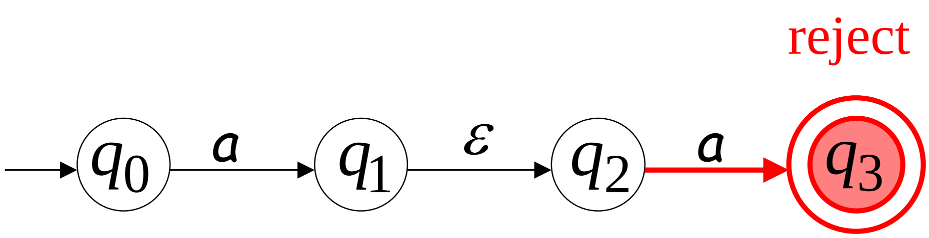
32


a
a
a
read head doesn't move
q_0
a
q_1
ε
q_2
a
q_3
33

Input cannot be consumed



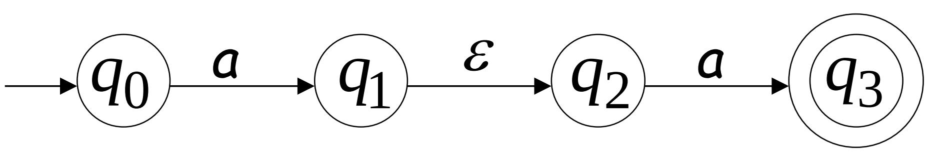
Automaton halts



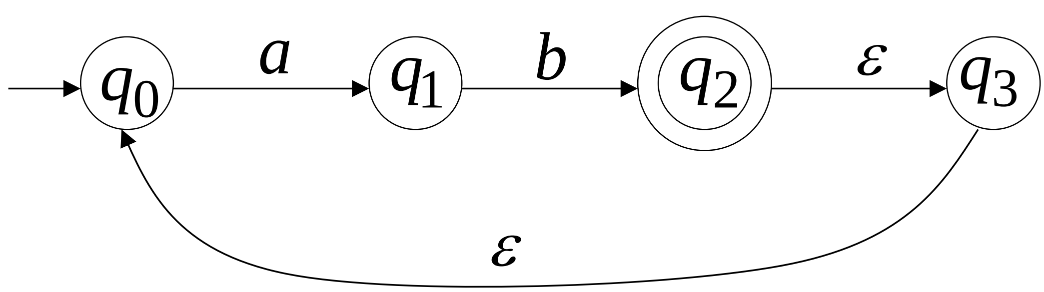
String aaa is rejected

34

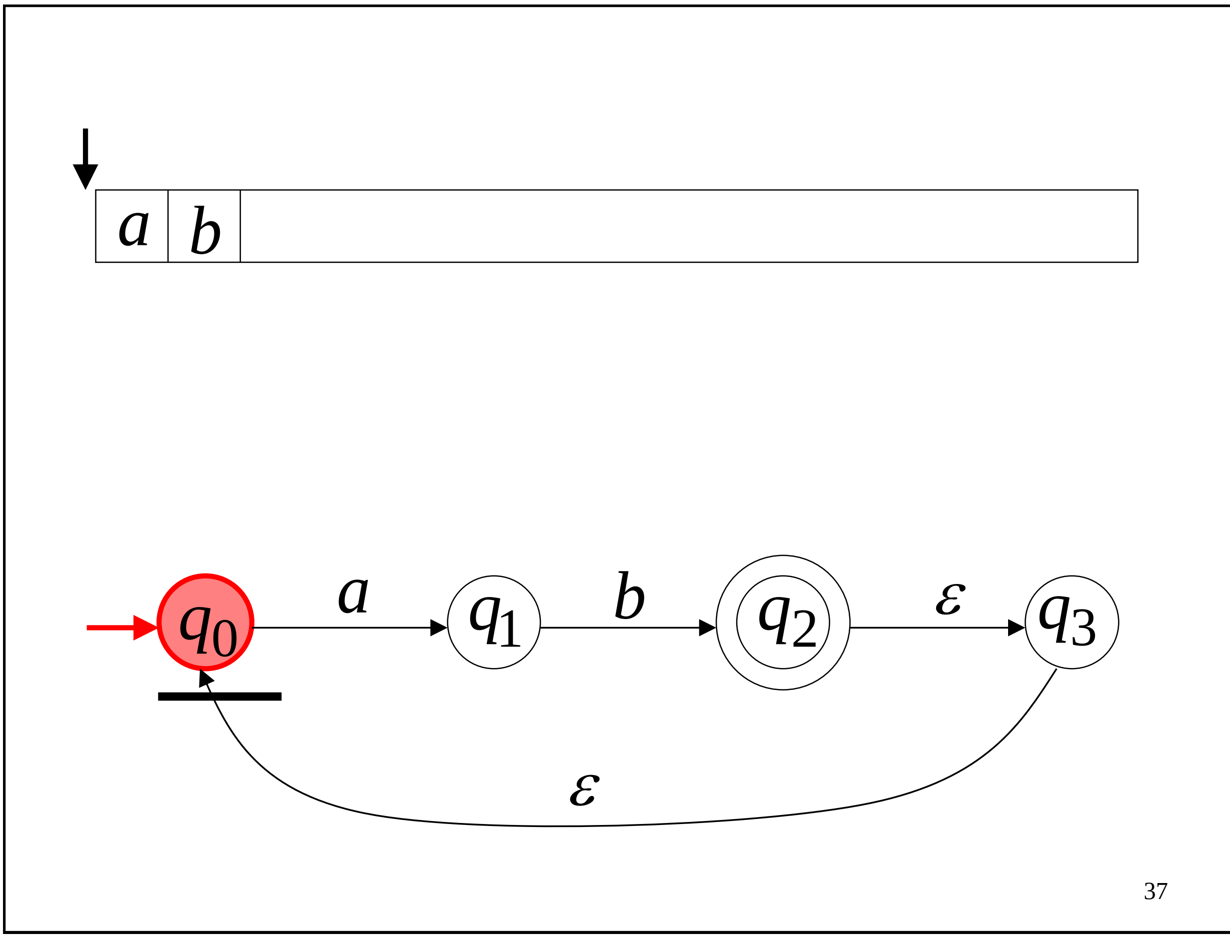
Language accepted: $L = \{aa\}$

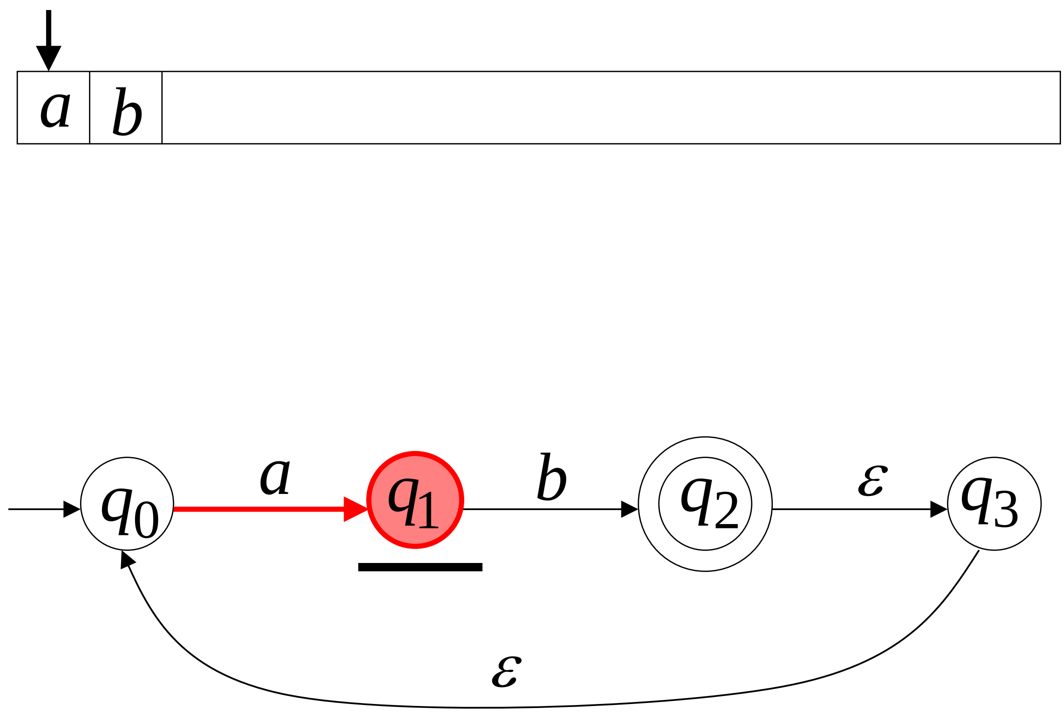


Another NFA Example

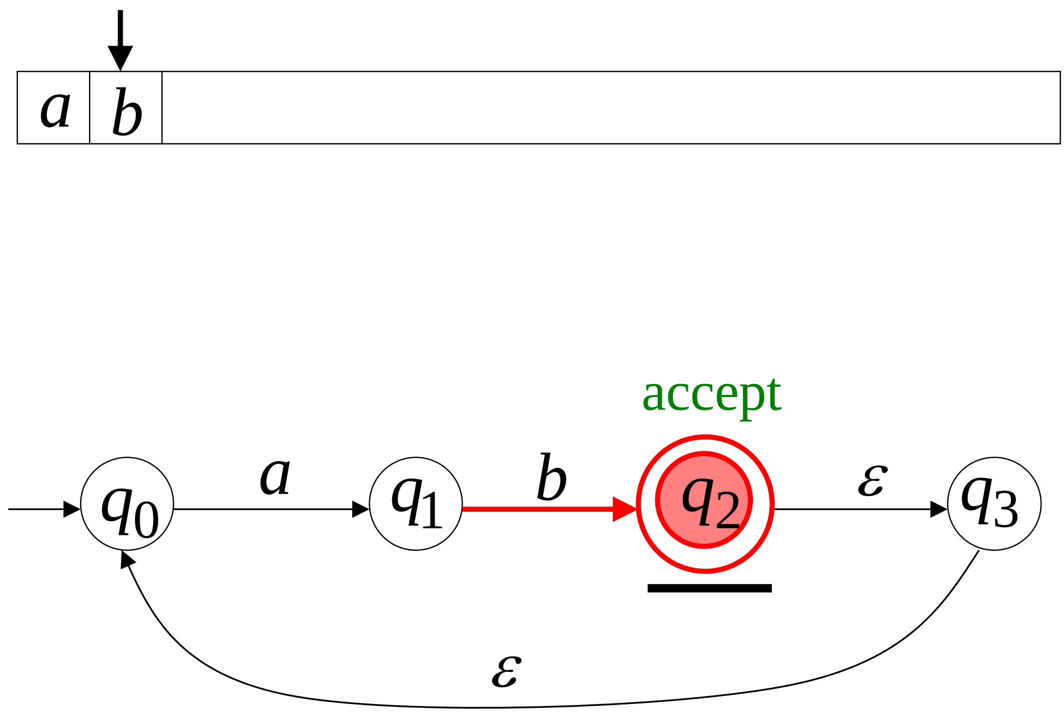


36


a
b
a
b
ε
ε
q0
q1
q2
q3
37

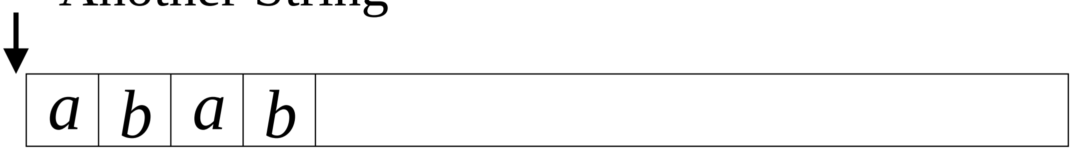


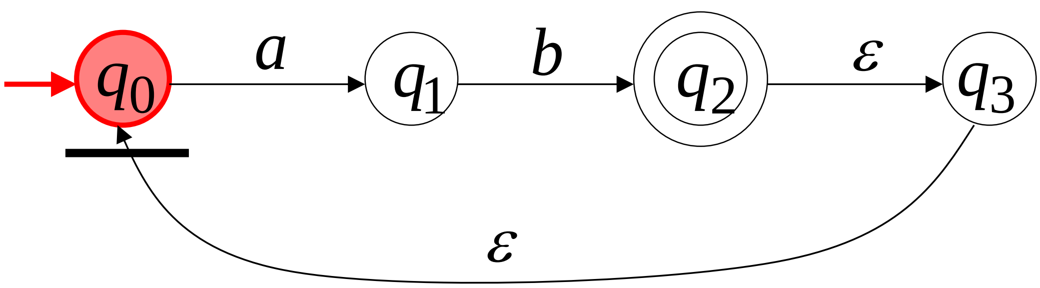
38

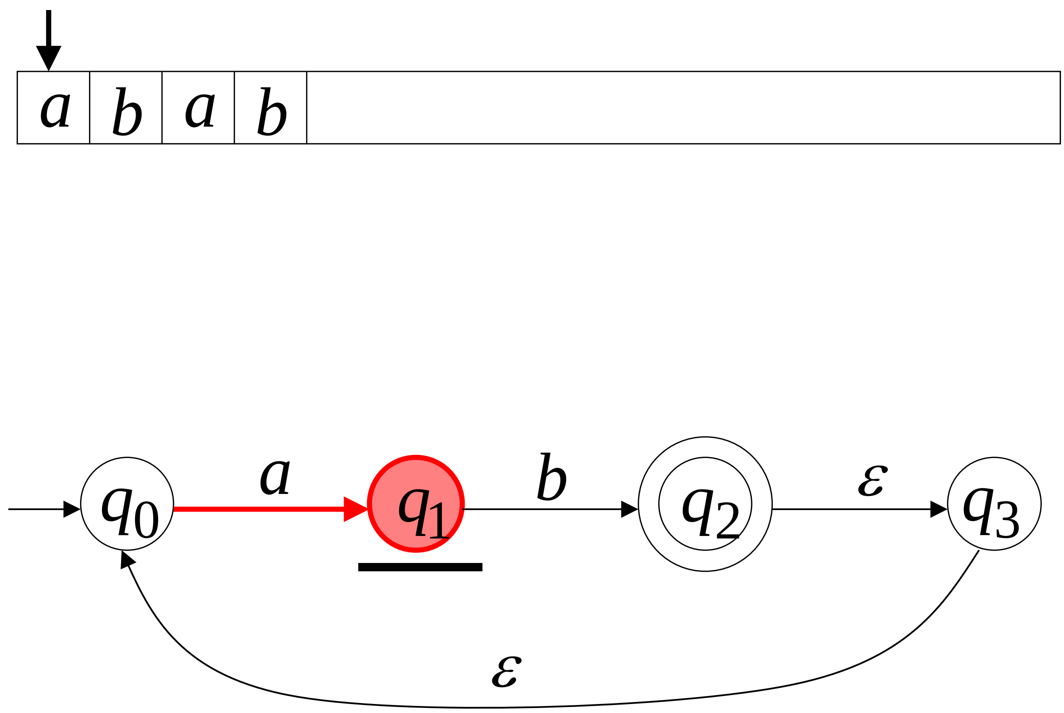


39

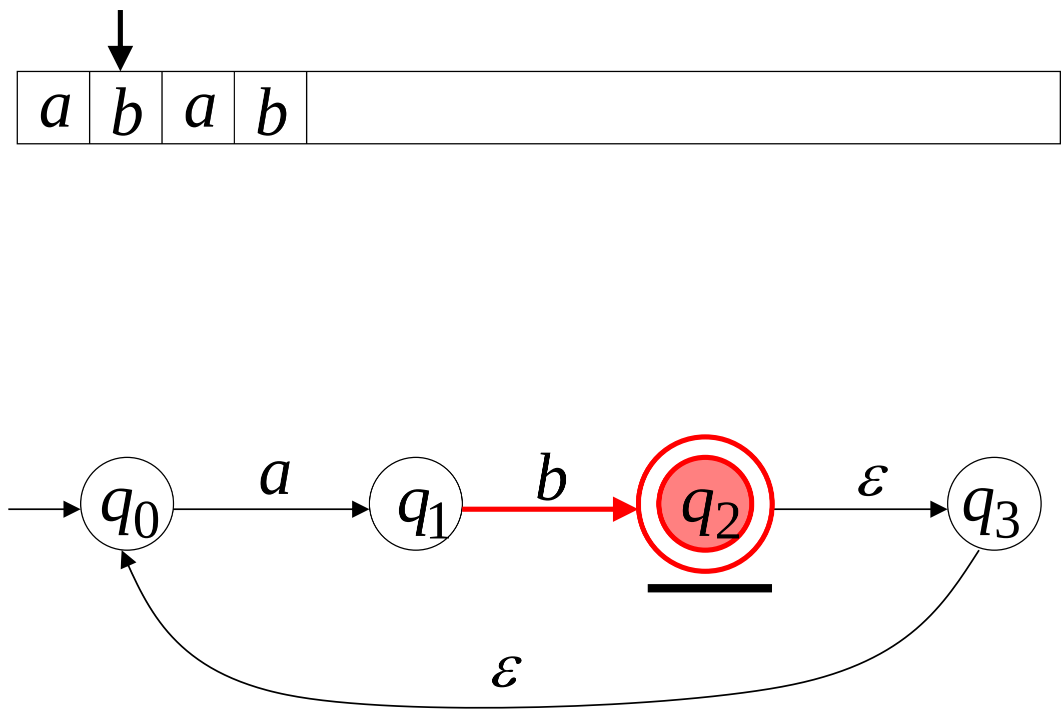
Another String



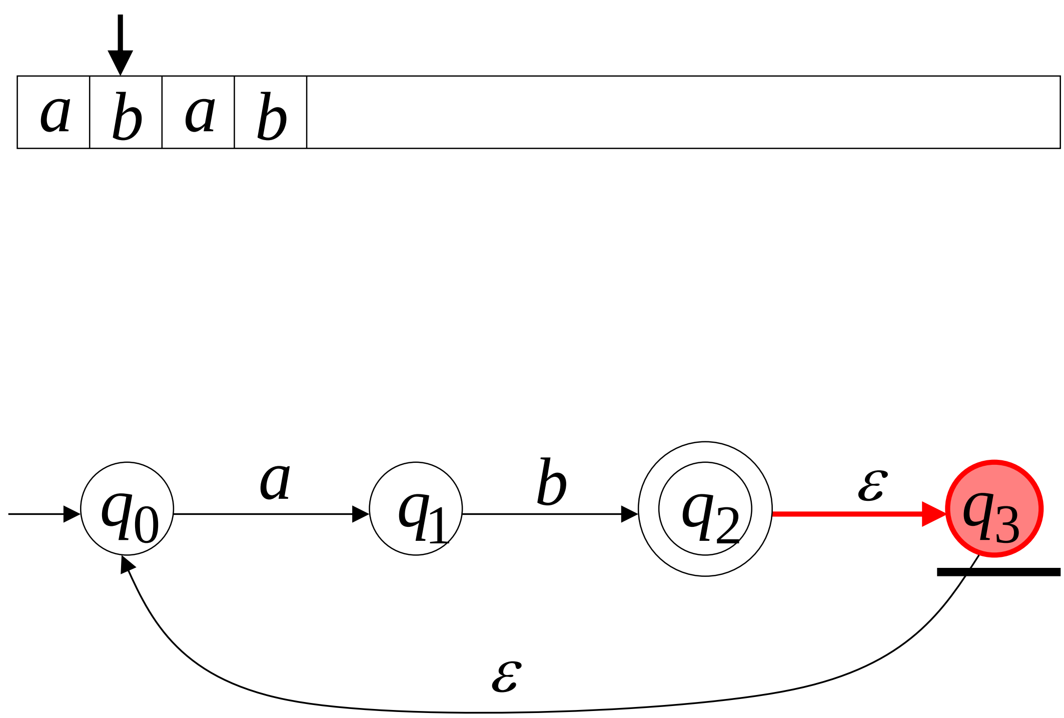




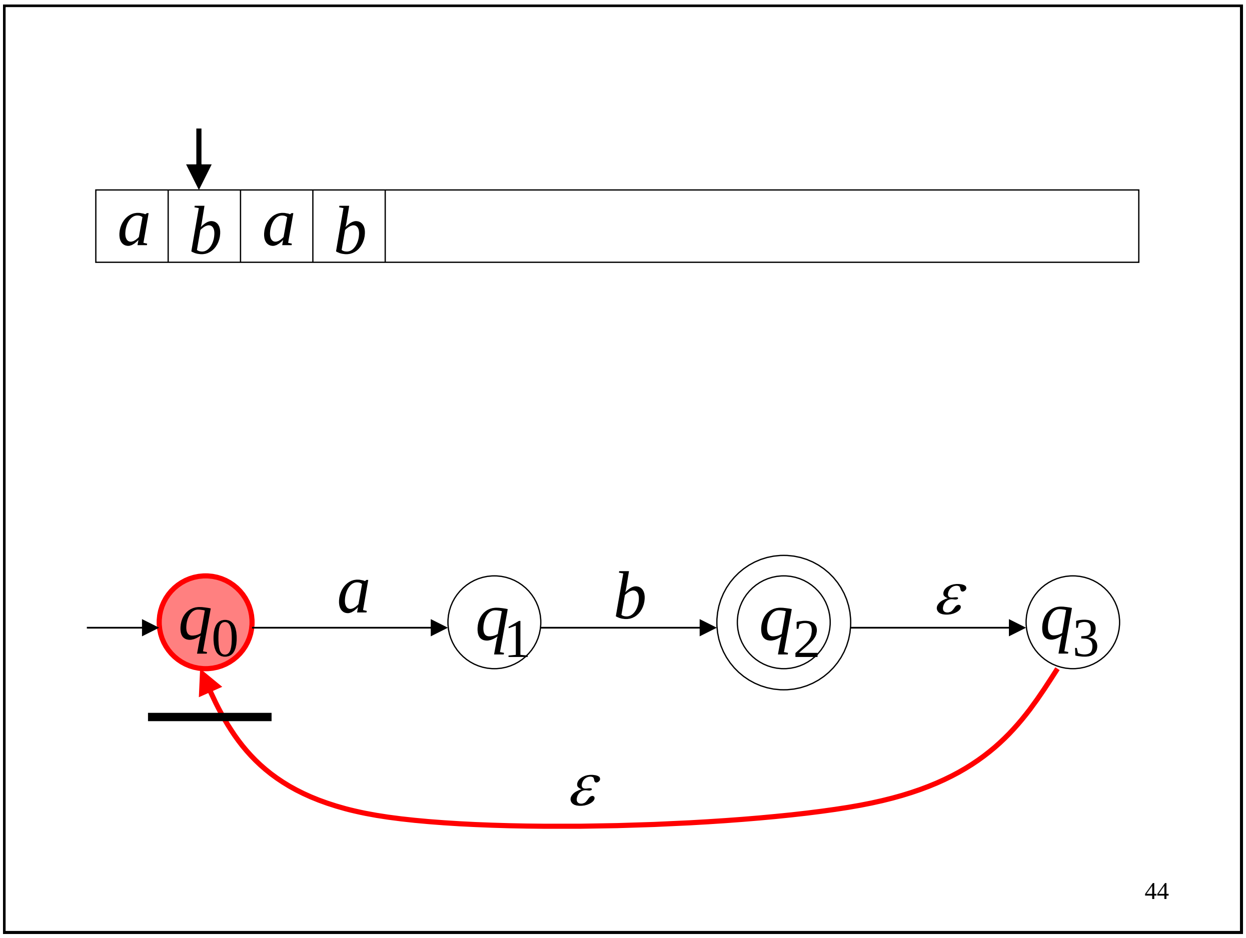
41



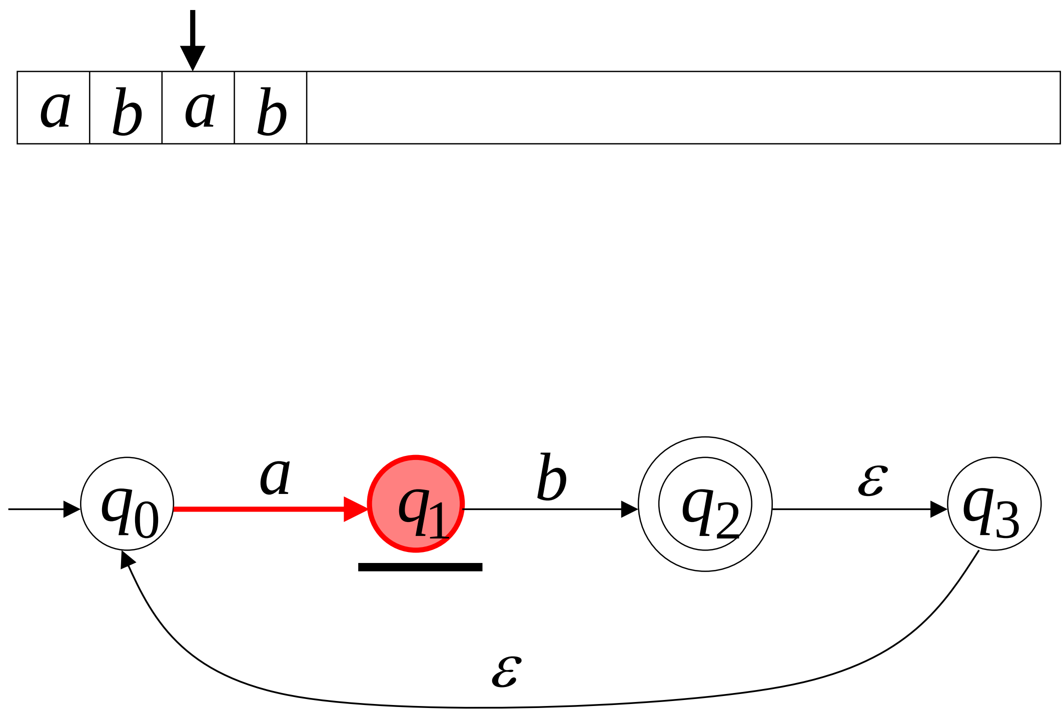
42



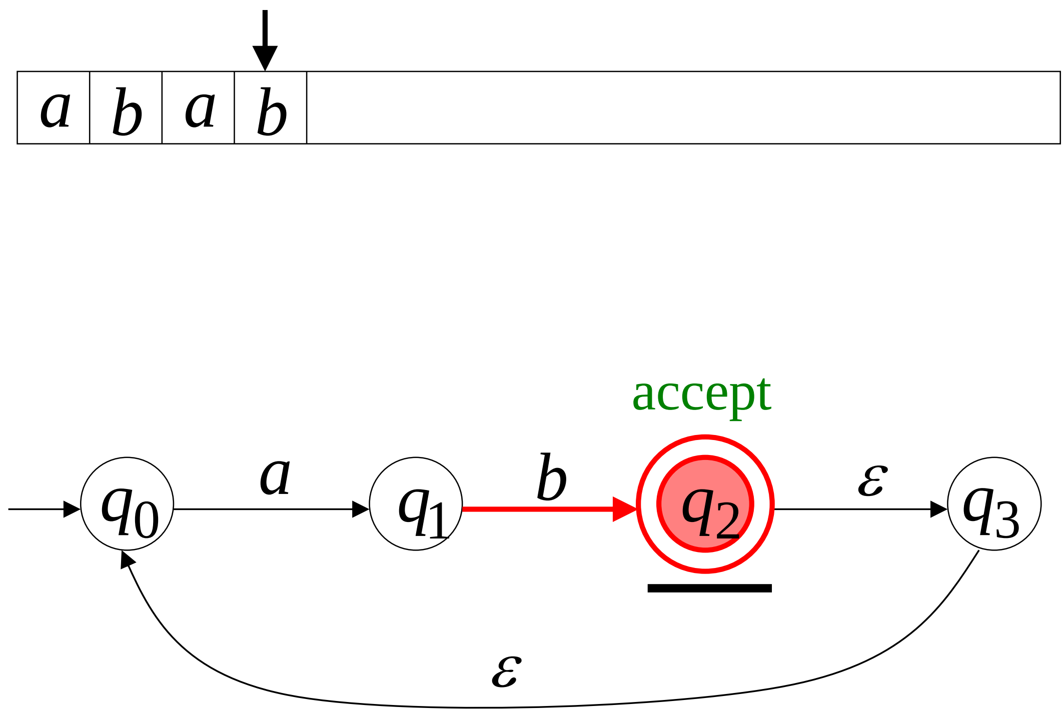
43



a b a b
a
b
ε
ε
q_0
q_1
q_2
q_3
44



45

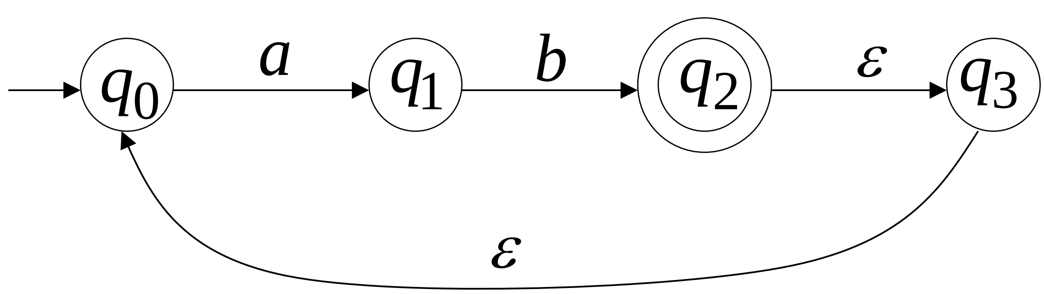


46

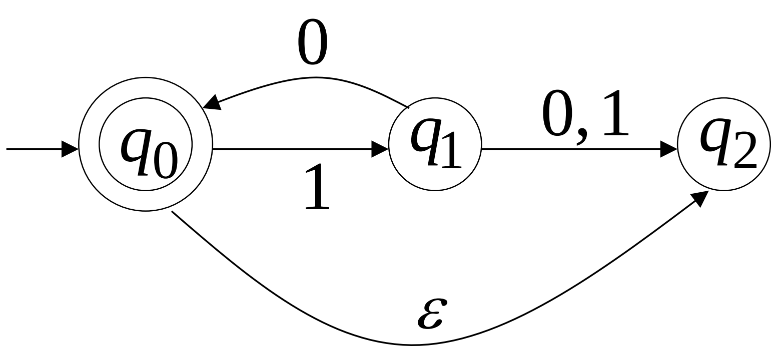
Language accepted

$$L = \{ab,\ abab,\ ababab,\ \ldots\}$$

$$= \{ab\}^+$$

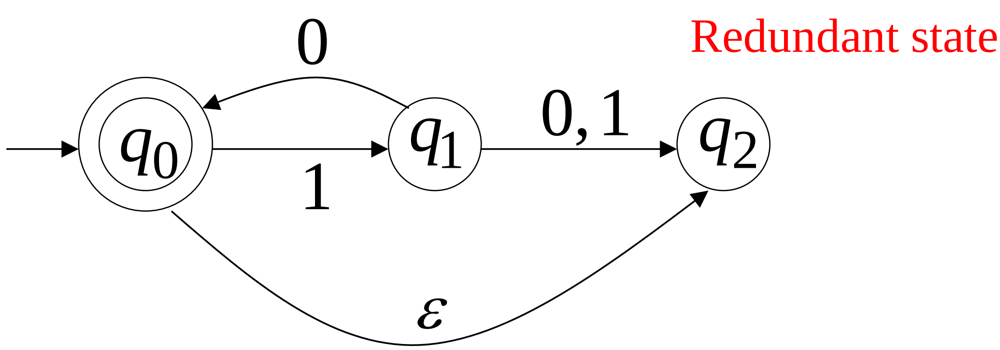


Another NFA Example



Language accepted

$$L(M) = \{\varepsilon,\ 10,\ 1010,\ 101010,\ \ldots\}$$
$$= \{10\}^*$$



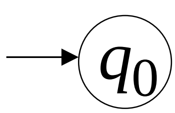
49

Remarks:

ε symbol never appears on the input tape.

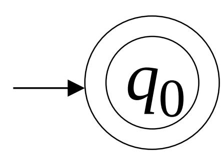
Simple automata:

M_1



$L(M_1) = \{\}$

M_2



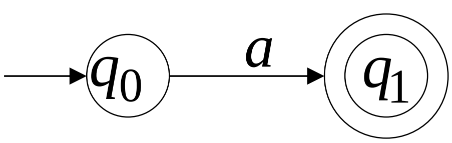
$L(M_2) = \{\varepsilon\}$

50

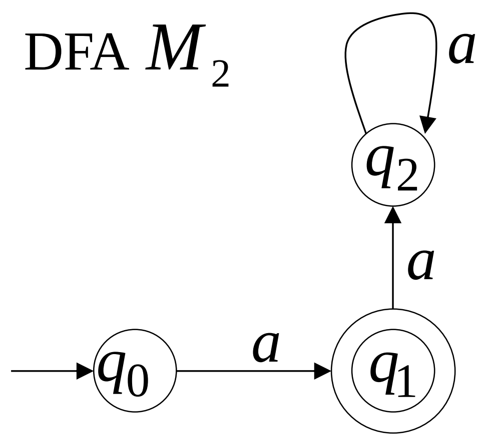
NFA vs. DFA

NFAs are interesting because we can express languages easier than DFAs

NFA M_1



DFA M_2



$L(M_1) = \{a\}$

$L(M_2) = \{a\}$

Formal Definition of NFAs

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

Q : Set of states, i.e. $\{q_0, q_1, q_2, \ldots\}$

Σ : Input alphabet, i.e. $\{a, b\}$

δ : Transition function

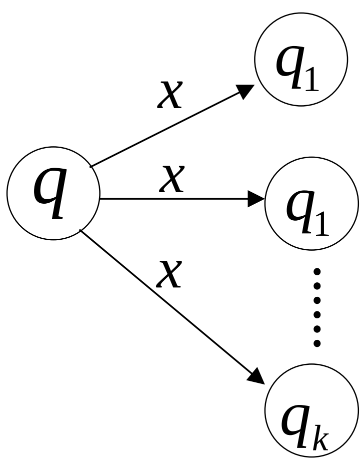
q_0 : Initial state

F : Accepting states

52

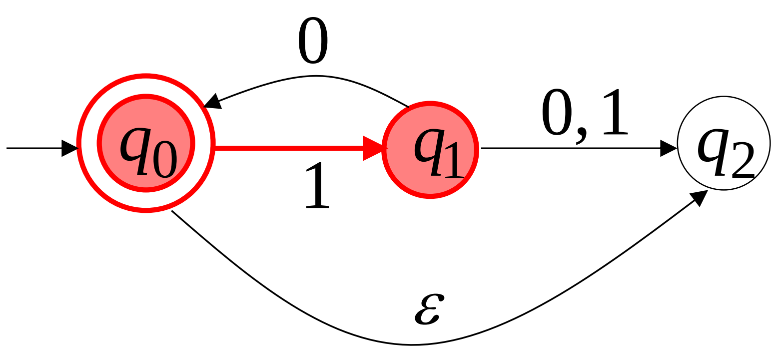
Transition Function δ

$$\delta(q, x) = \{q_1, q_2, \ldots, q_k\}$$



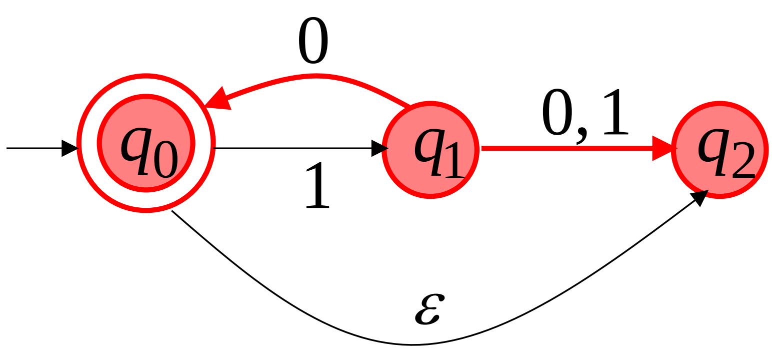
Resulting states with following one transition with symbol x

$$\delta(q_0, 1) = \{q_1\}$$



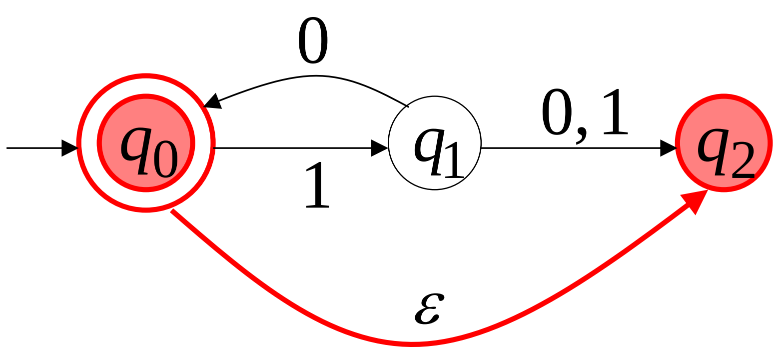
54

$$\delta(q_1,0) = \{q_0, q_2\}$$



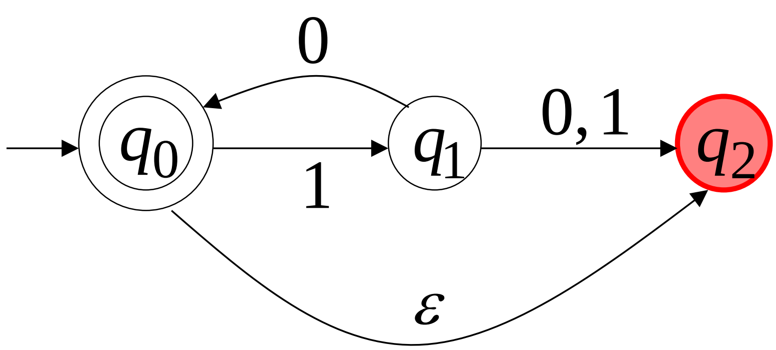
55

$$\delta(q_0, \lambda) = \{q_2\}$$



56

$$\delta(q_2, 1) = \varnothing$$

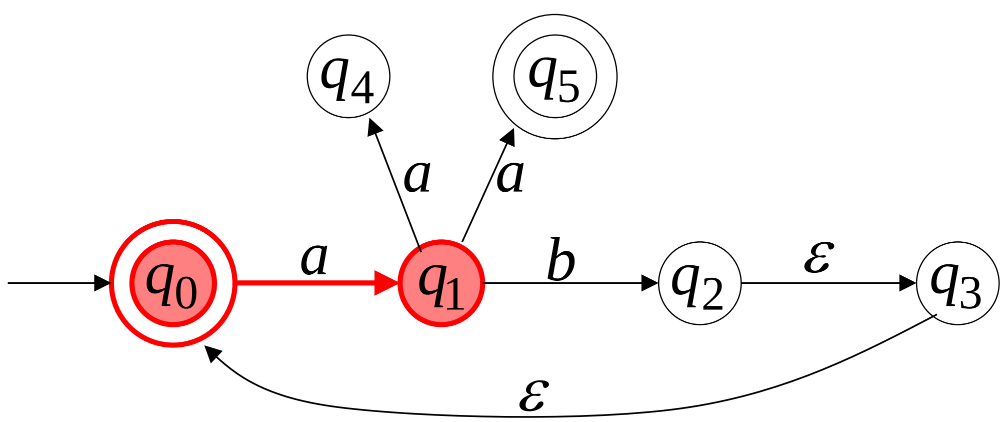


57

Extended Transition Function δ^*

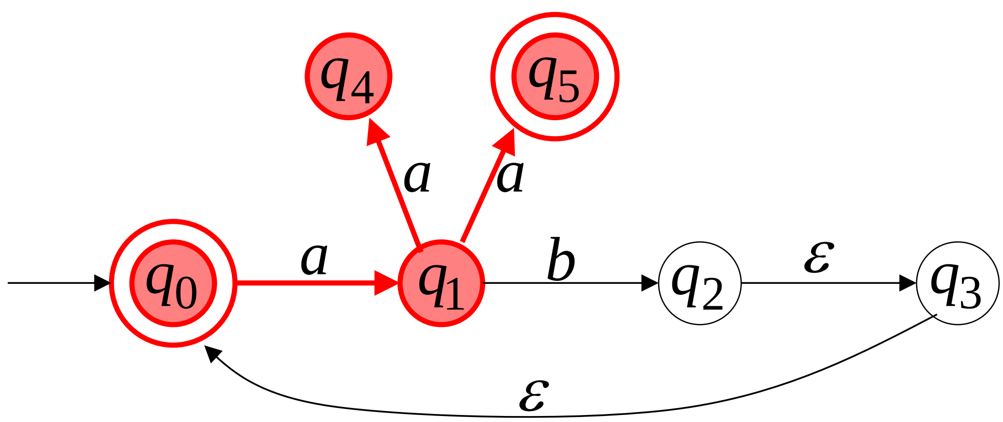
Same with δ but applied on strings

$$\delta^*(q_0, a) = \{q_1\}$$



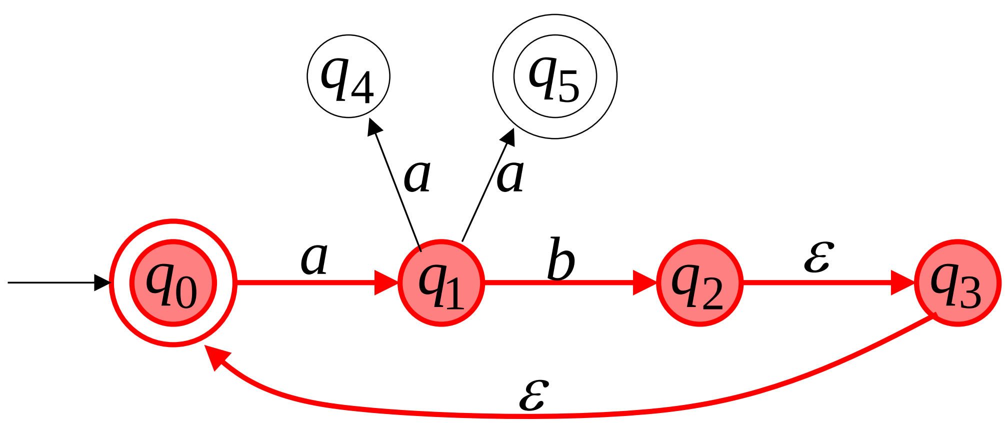
58

$$\delta^*(q_0, aa) = \{q_4, q_5\}$$



59

$$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$$



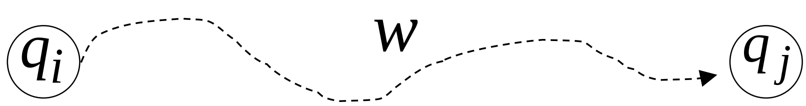
60

Particular case:

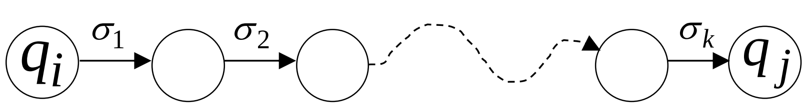
for any state q, $q \in \delta^*(q, \varepsilon)$

In general

$q_j \in \delta^*(q_i, w)$: there is a walk from q_i to q_j with label w



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



The Language of an NFA M

- The language accepted by M is:

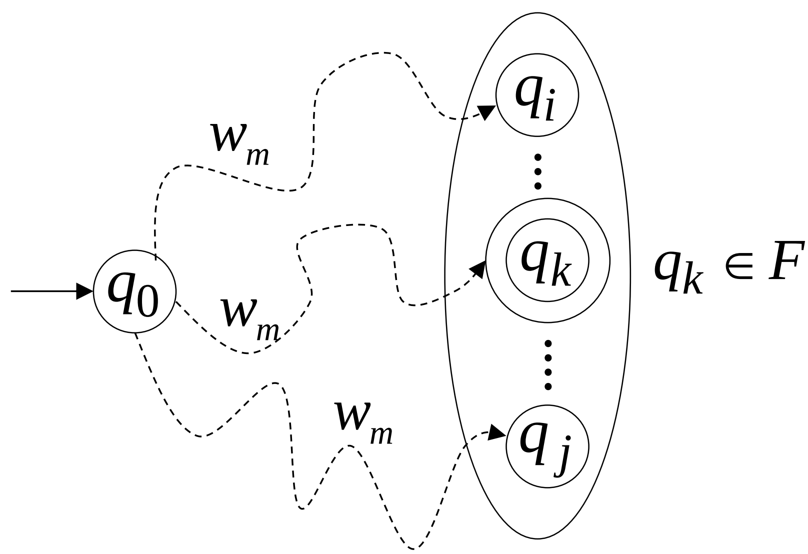
$$L(M) = \{w_1, w_2, \ldots w_n\}$$

where $\delta^*(q_0, w_m) = \{q_i, \ldots, q_k, \ldots, q_j\}$

and there is some $q_k \in F$ (accepting state)

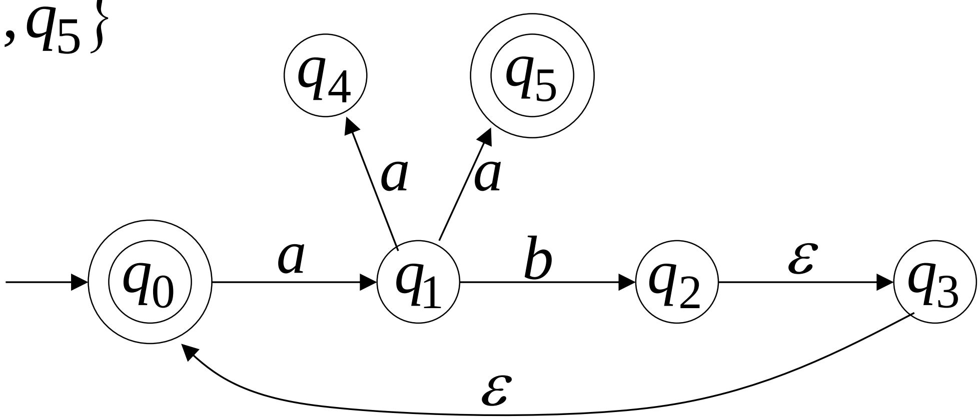
$w_m \in L(M)$

$\delta^*(q_0, w_m)$



64

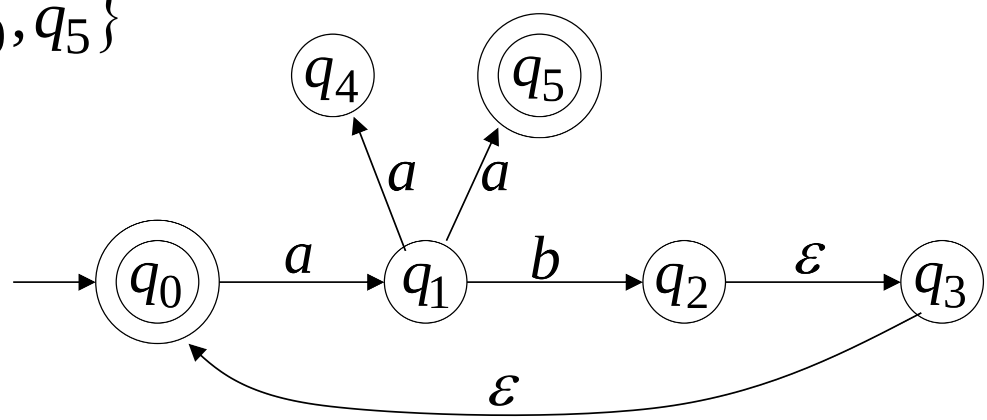
$$F = \{q_0, q_5\}$$



$$\delta^*(q_0, aa) = \{q_4, \underbrace{q_5}_{\in F}\} \Longrightarrow aa \in L(M)$$

65

$$F=\{q_0,q_5\}$$

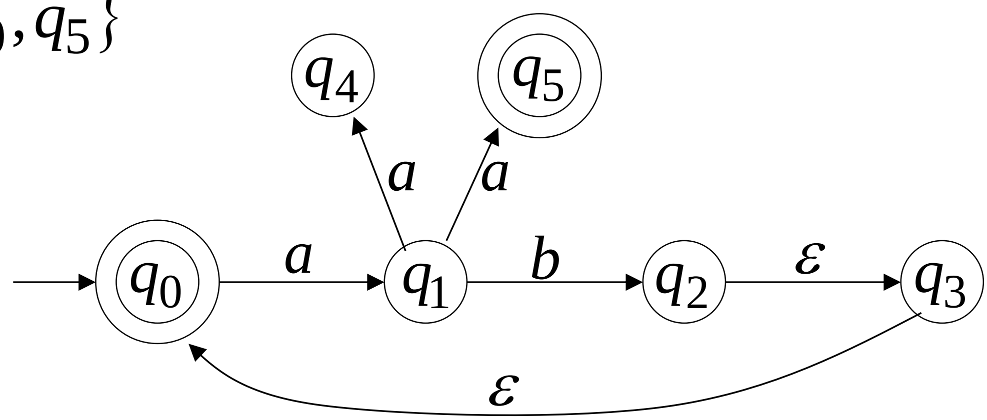


$$\delta^*(q_0,ab)=\{q_2,q_3,\underline{q_0}\} \Rightarrow ab\in L(M)$$

$$\in F$$

66

$$F = \{q_0, q_5\}$$

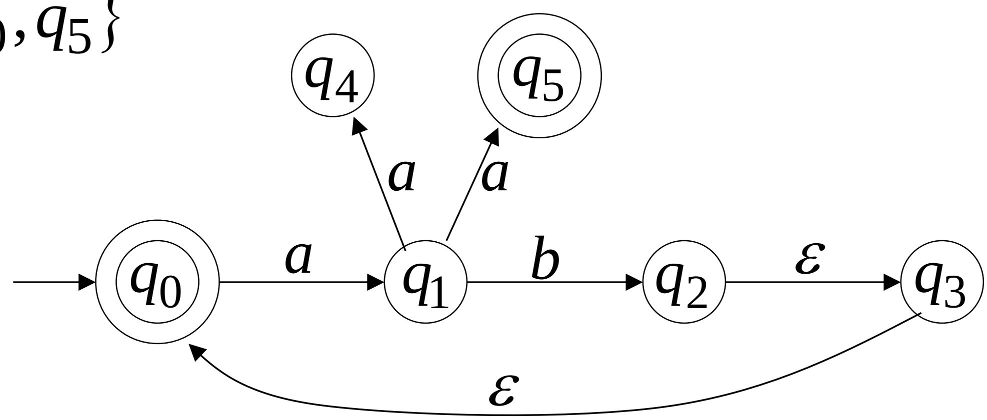


$$\delta^*(q_0, abaa) = \{q_4, \underline{q_5}\} \Longrightarrow abaa \in L(M)$$

$$\in F$$

67

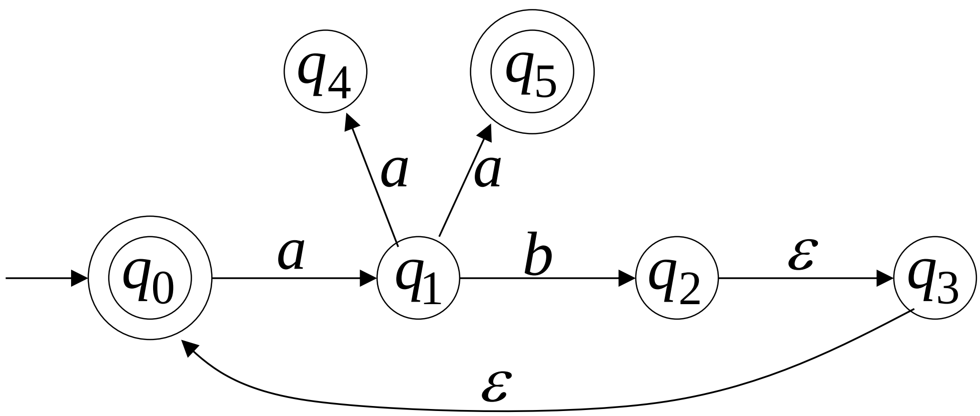
$$F = \{q_0, q_5\}$$



$$\delta^*(q_0, aba) = \{q_1\} \quad \Rightarrow \quad aba \notin L(M)$$

$$\notin F$$

68



$$L(M) = \{ab\}^* \cup \{ab\}^*\{aa\}$$

69

NFAs accept the Regular Languages

Equivalence of Machines

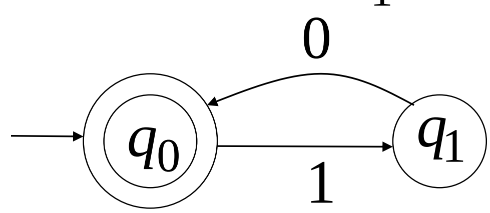
Definition:

- The machine M_1 is equivalent to the machine M_2 if $L(M_1) = L(M_2)$

Example of equivalent machines

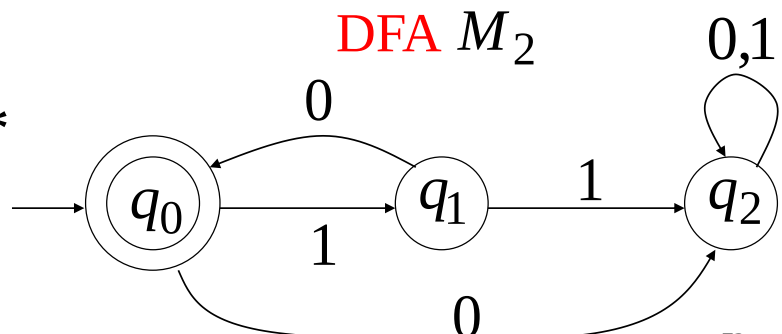
NFA M_1

$$L(M_1) = \{10\}^*$$



DFA M_2

$$L(M_2) = \{10\}^*$$



Languages accepted by NFA and DFA

Theorem:

$$\left\{\begin{array}{l}\text{Languages}\\\text{accepted}\\\text{by NFAs}\end{array}\right\} = \left\{\begin{array}{l}\text{Regular}\\\text{Languages}\end{array}\right\}$$

Languages
accepted
by DFAs

NFAs and DFAs have the same computation power.
They accept the same set of languages.

Languages accepted by NFA and DFA

Proof: We need to show:

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

AND

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

Languages accepted by NFA and DFA

Step 1

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

Every DFA is trivially an NFA

Then

Any language L accepted by a DFA is also accepted by an NFA

Languages accepted by NFA and DFA

Step 2

$$\left\{ \begin{array}{l} \text{Languages} \\ \text{accepted} \\ \text{by NFAs} \end{array} \right\} \subseteq \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$$

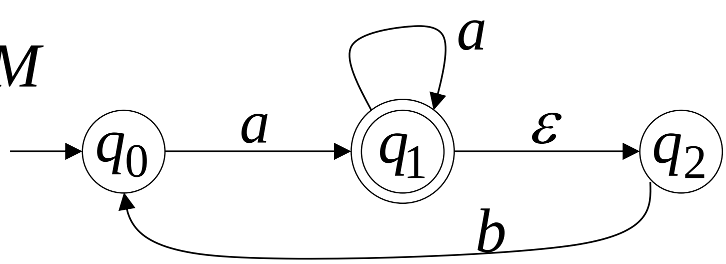
Any NFA can be converted to an equivalent DFA

Then

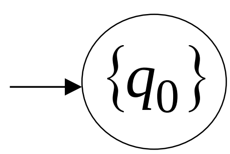
Any language L accepted by an NFA is also accepted by a DFA

Conversion of NFA to DFA

NFA M



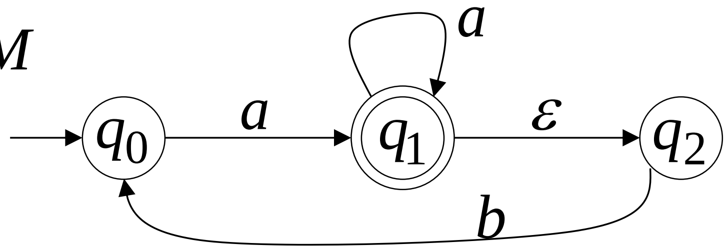
DFA M'



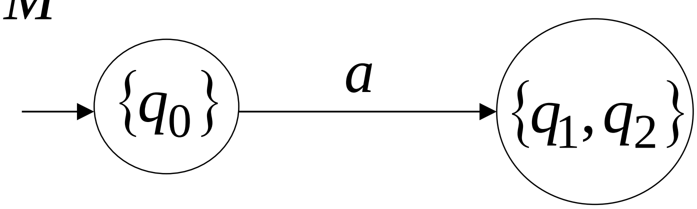
77

$$\delta^*(q_0, a) = \{q_1, q_2\}$$

NFA M

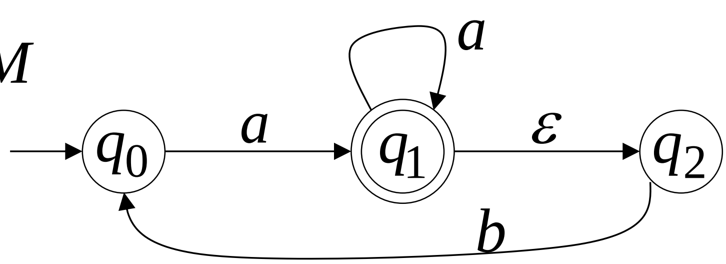


DFA M'

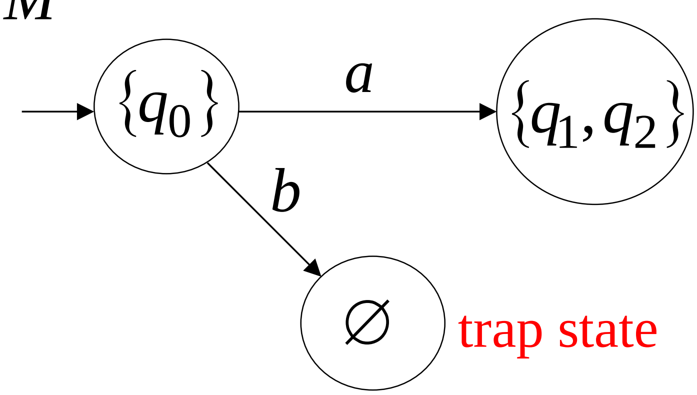


$$\delta^*(q_0, b) = \varnothing$$

NFA M

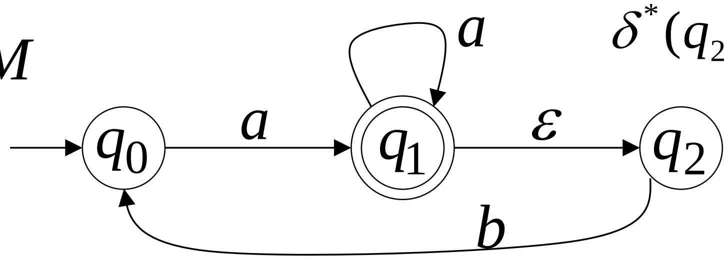


DFA M'



79

NFA M

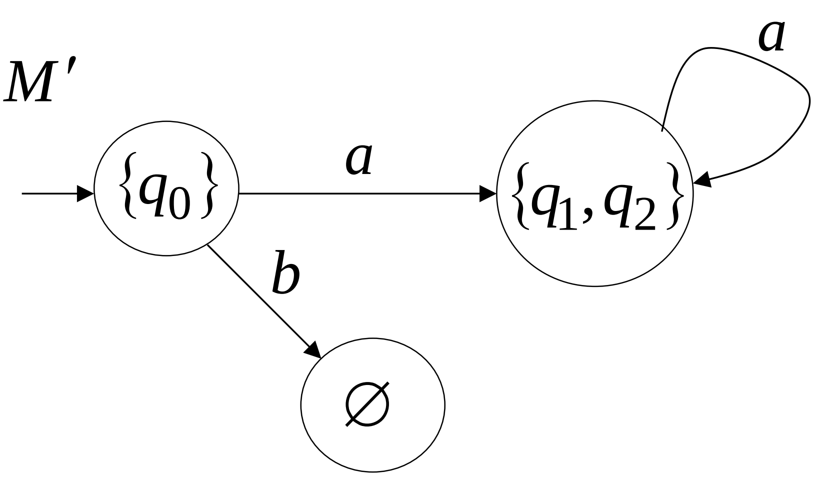


$$\delta^*(q_1, a) = \{q_1, q_2\}$$
$$\delta^*(q_2, a) = \varnothing$$

union

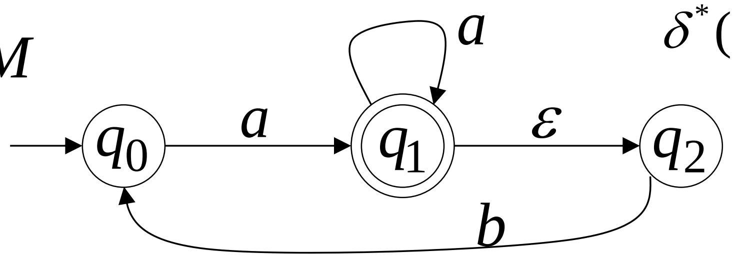
$\{q_1, q_2\}$

DFA M'



80

NFA M

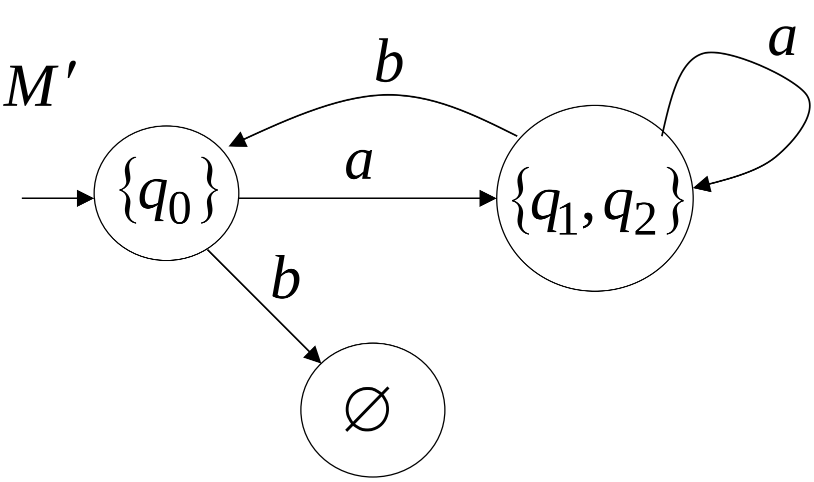


$$\delta^*(q_1, b) = \{q_0\}$$
$$\delta^*(q_2, b) = \{q_0\}$$

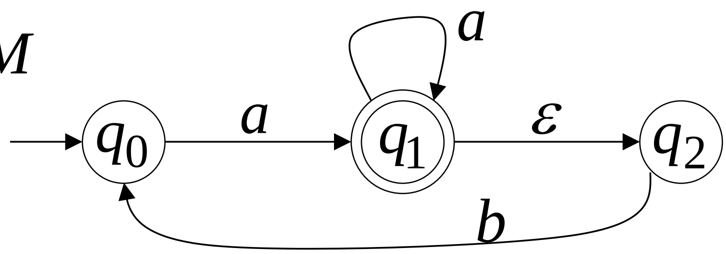
union

$\{q_0\}$

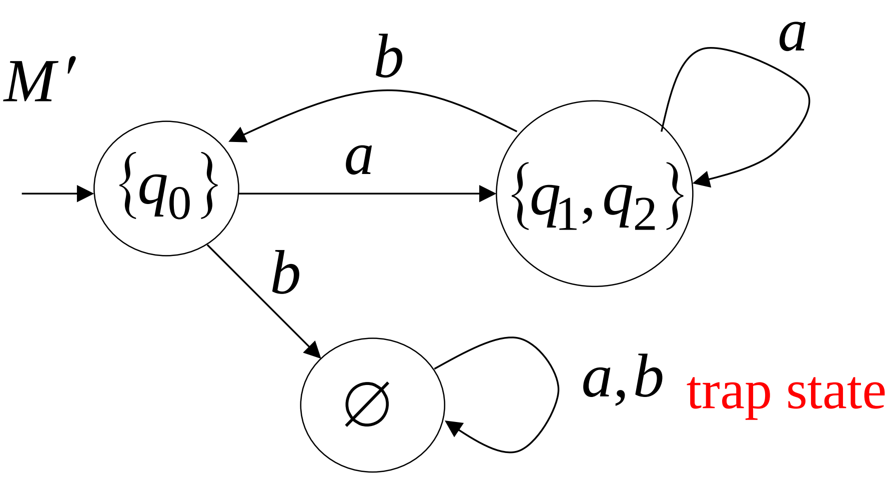
DFA M'



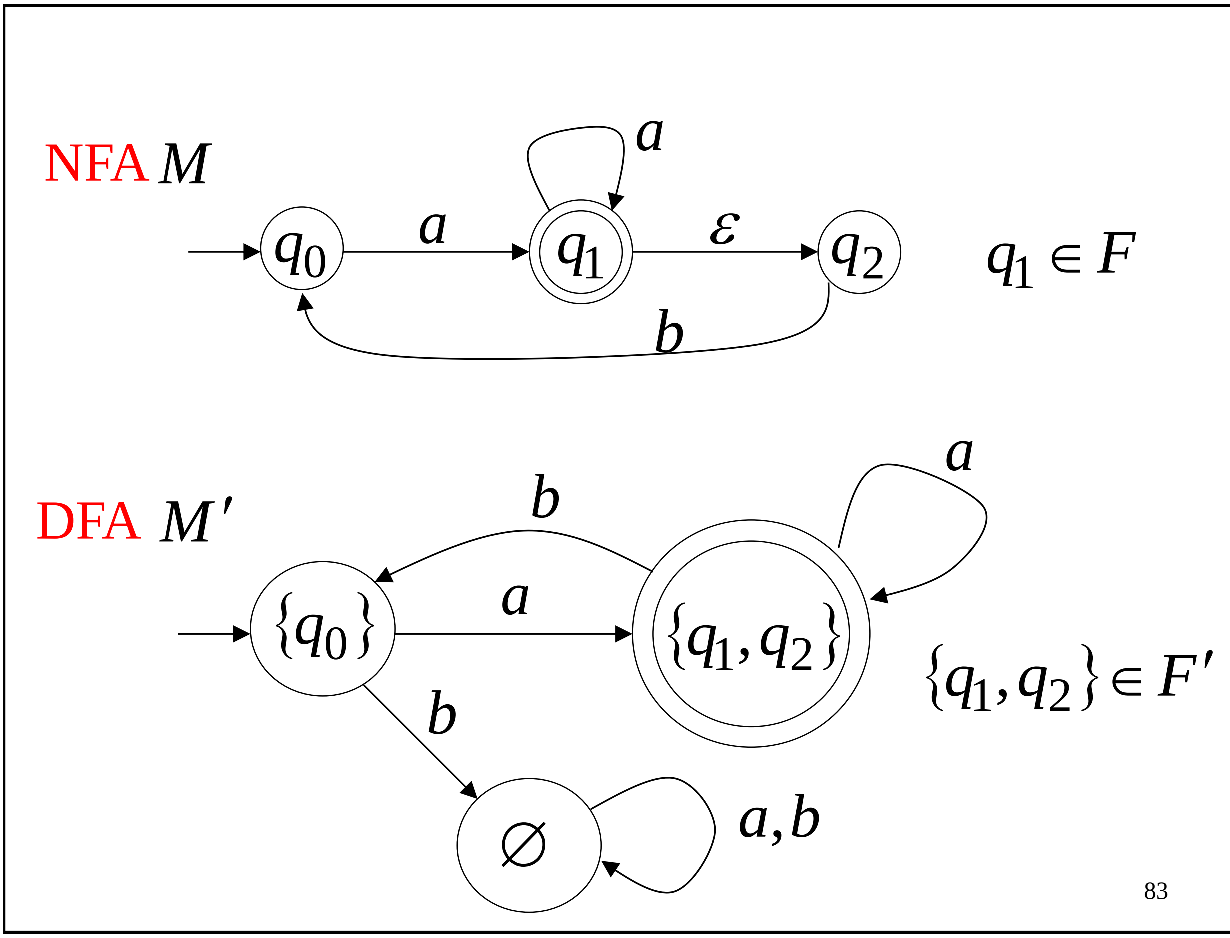
NFA M



DFA M'



82


NFA M
q_0
a
q_1
a
ε
q_2
b
$q_1 \in F$
DFA M'
$\{q_0\}$
a
b
$\{q_1, q_2\}$
a
b
$\emptyset$
a, b
$\{q_1, q_2\} \in F'$
83

General Conversion Procedure

- Input: an NFA M
- Output: an equivalent DFA M'
  with $L(M) = L(M')$

General Conversion Procedure

- The NFA has states $q_0, q_1, q_2, ...$

- The DFA has states from the power set:

$$\emptyset, \{q_0\}, \{q_1\}, \{q_0, q_1\}, \{q_1, q_2, q_3\}, ...$$

General Conversion Procedure

1. Initial state of NFA: q_0

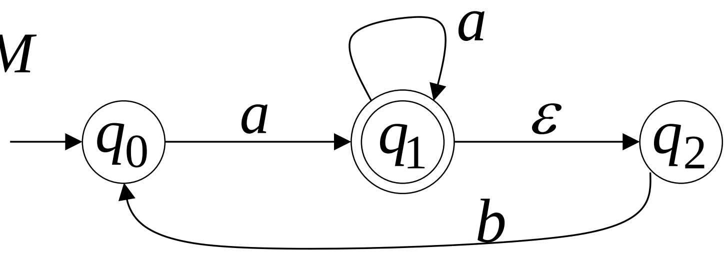
$$\Downarrow$$

Initial state of DFA: $\{q_0\}$

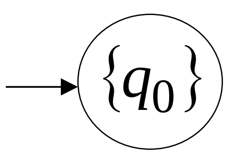
86

Example:

NFA M



DFA M'



87

2. For every DFA's state $\{q_i, q_j, \ldots, q_m\}$

compute in the NFA

$$\left.\begin{array}{c} \delta^*(q_i, a) \\ \cup\, \delta^*(q_j, a) \\ \ldots \\ \cup\, \delta^*(q_m, a) \end{array}\right\} \overset{\textbf{Union}}{=} \{q'_k, q'_l, \ldots, q'_n\}$$

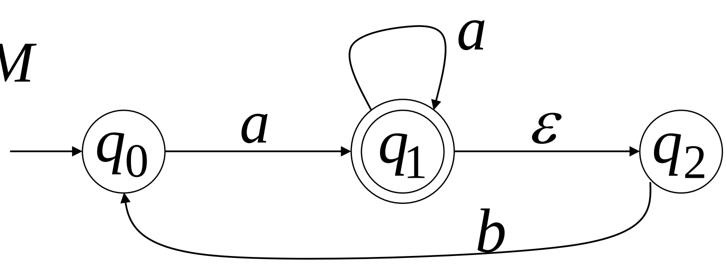
add transition to DFA

$$\delta(\{q_i, q_j, \ldots, q_m\},\ a) = \{q'_k, q'_l, \ldots, q'_n\}$$

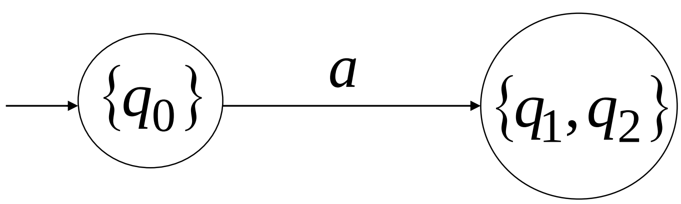
88

Example $\delta^*(q_0, a) = \{q_1, q_2\}$

NFA M



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



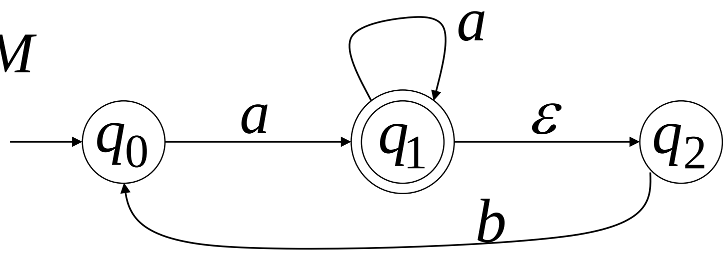
89

3. Repeat Step **2** for every state in DFA and symbols in alphabet until no more states can be added in the DFA.

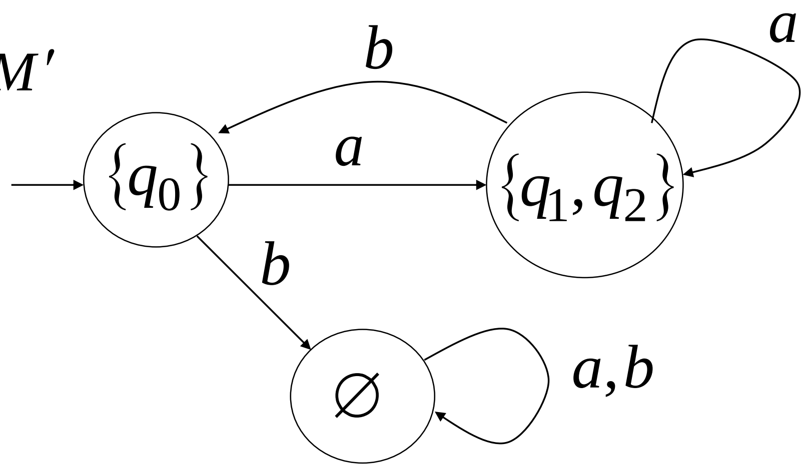
90

Example

NFA M



DFA M'



91

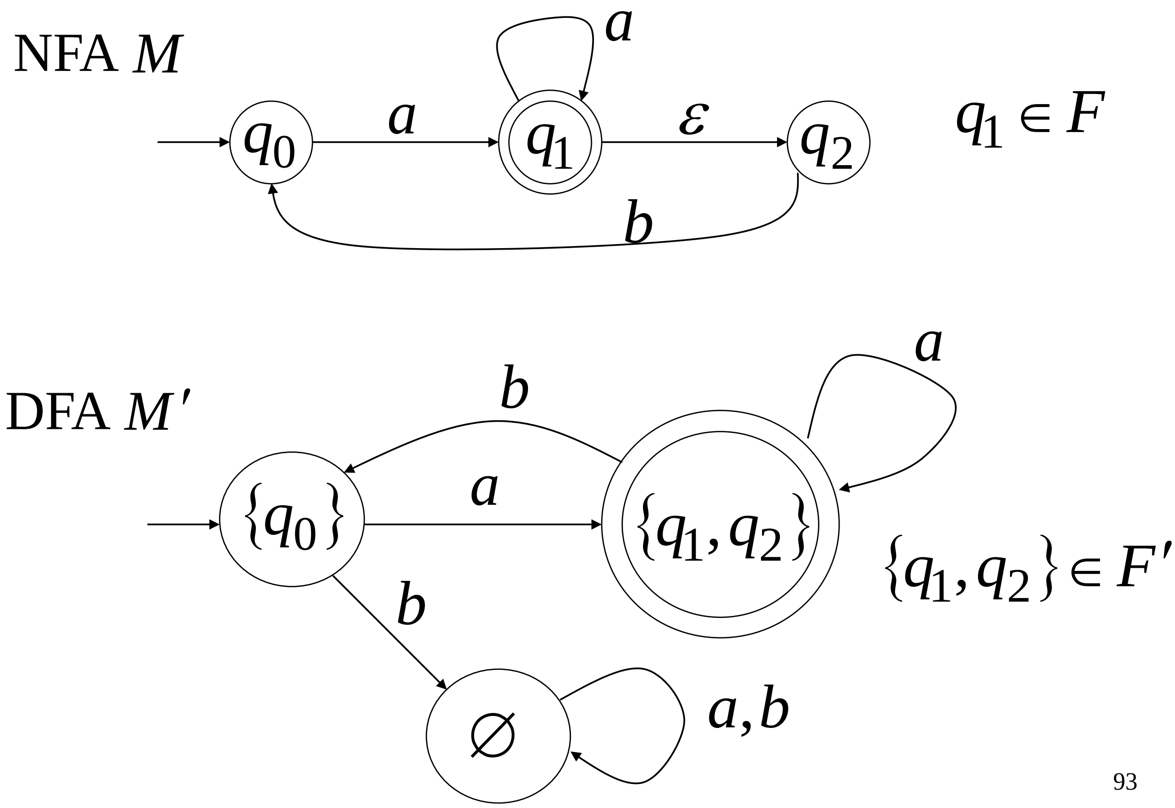
4. For any DFA state $\{q_i, q_j, ..., q_m\}$

If some q_j is accepting state in NFA

Then

$\{q_i, q_j, ..., q_m\}$ is accepting state in DFA

Example


NFA M
q_0
a
q_1
a
ε
q_2
b
$q_1 \in F$
DFA M'
$\{q_0\}$
a
b
$\{q_1, q_2\}$
a
b
$\emptyset$
a, b
$\{q_1, q_2\} \in F'$

93

NFA and DFA are Equivalent

Lemma:

If we convert NFA M to DFA M'

then the two automata are equivalent:

$$L(M) = L(M')$$

Proof:

We need to show: $L(M) \subseteq L(M')$

AND

$$L(M) \supseteq L(M')$$

94

First we show: $L(M) \subseteq L(M')$

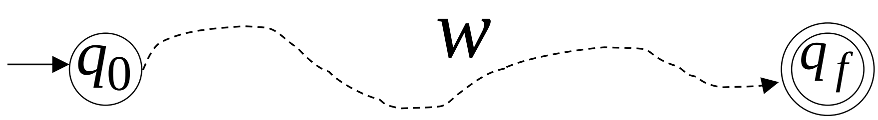
We need to prove:

$w \in L(M)$ **Then** $w \in L(M')$

95

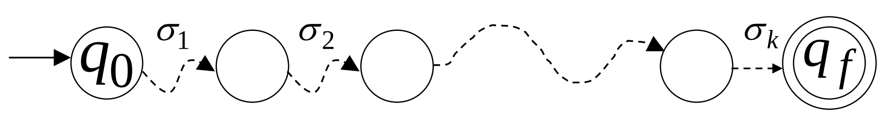
Let’s have an NFA

Consider $w \in L(M)$

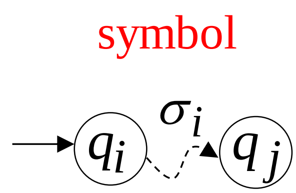


symbols

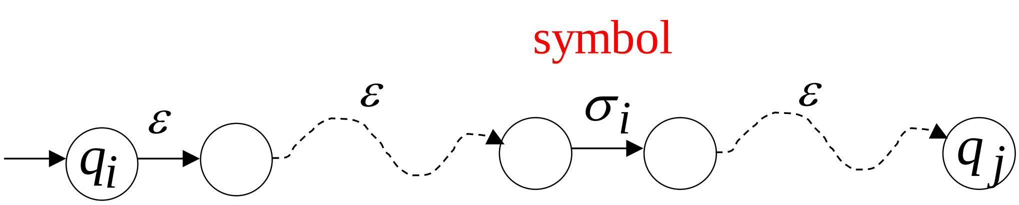
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



96



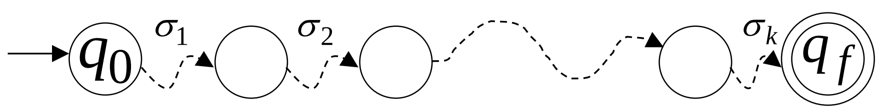
denotes a possible sub-path like



We show that if $w \in L(M)$

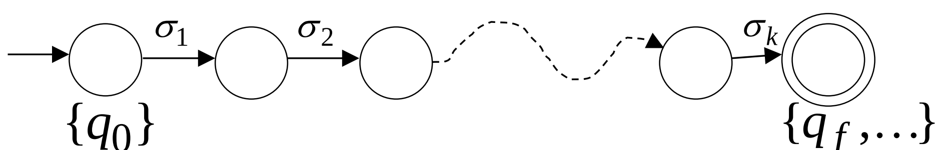
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

NFA $M:$



Then

DFA $M':$



$w \in L(M')$

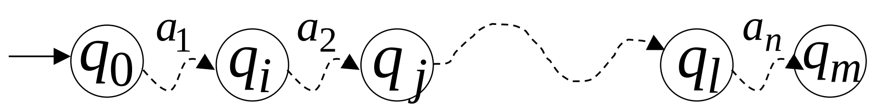
state label

state label

More generally, we show that if in M :

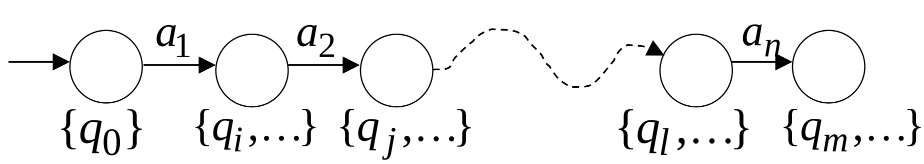
An arbitrary string $v = a_1 a_2 \cdots a_n$

NFA M :



Then

DFA M' :



99

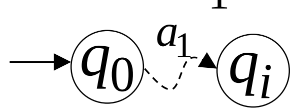
NFA and DFA are Equivalent

Proof by induction on $|v|$

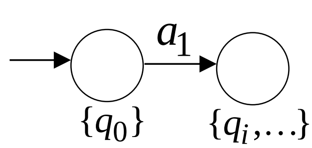
Induction Basis: $|v| = 1$

$v = a_1$

NFA M :



DFA M' :



is true by construction of M'

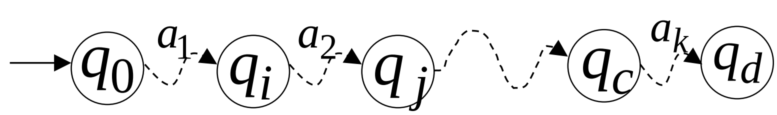
100

Induction hypothesis: $1 \leq |v| \leq k$

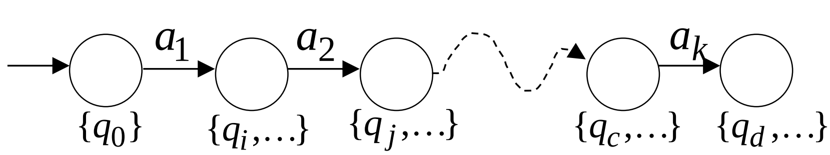
$$v = a_1 a_2 \cdots a_k$$

Suppose that the following hold

NFA M :



DFA M' :

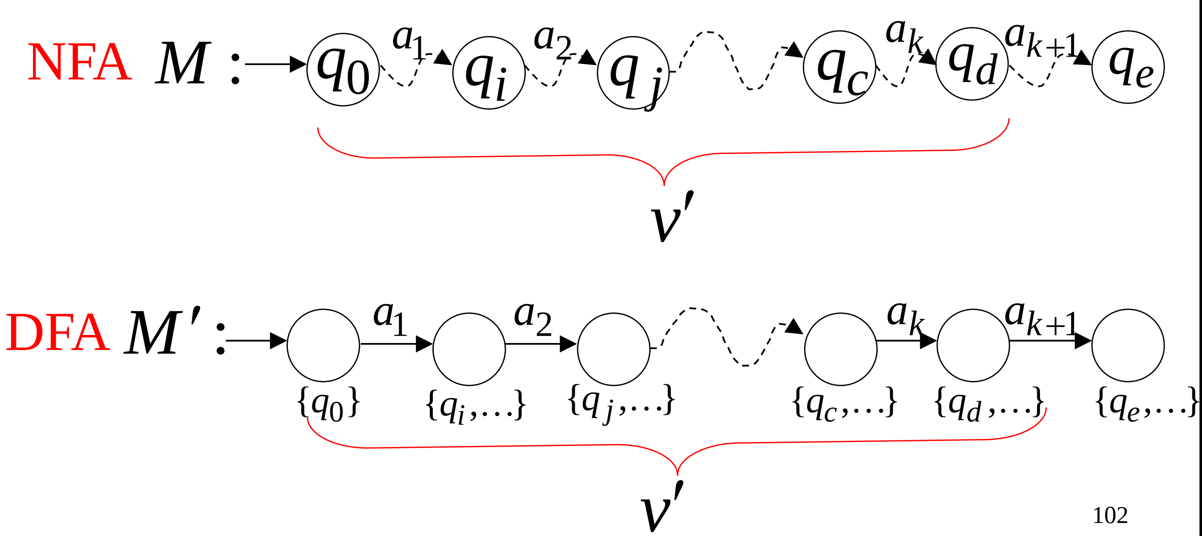


101

Induction Step: $|v| = k+1$

$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$

Then this is true by construction of M'

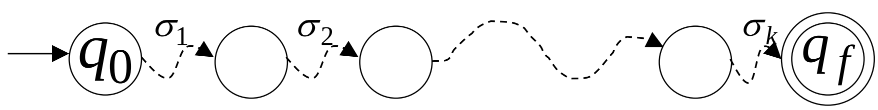


102

Therefore if $w \in L(M)$

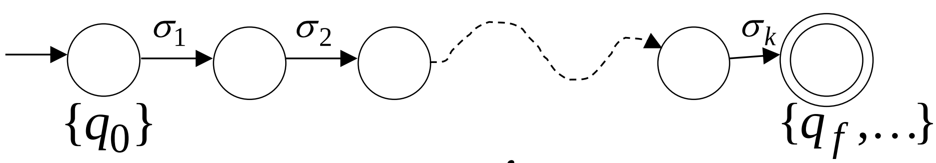
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

NFA M :



Then

DFA M' :



$$w \in L(M')$$

103

We have shown: $L(M) \subseteq L(M')$

With a similar proof, we can show: $L(M) \supseteq L(M')$

Therefore: $L(M) = L(M')$

104

Readings

- Textbook:
  - Part 1, Section 1.2 (Finite Automata)