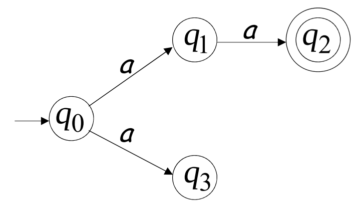
Non Deterministic Automata

Class 3

Nondeterministic Finite Accepter (NFA)

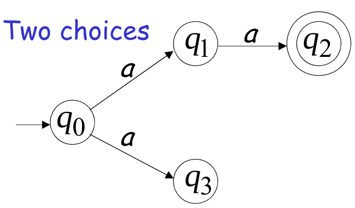
Alphabet = $\{a\}$



Nondeterministic Finite Accepter (NFA)

Alphabet = $\{a\}$

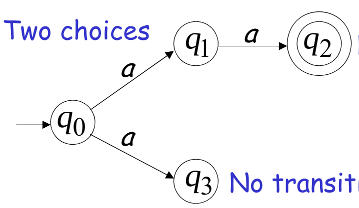
Two choices



Nondeterministic Finite Accepter (NFA)

Alphabet = $\{a\}$

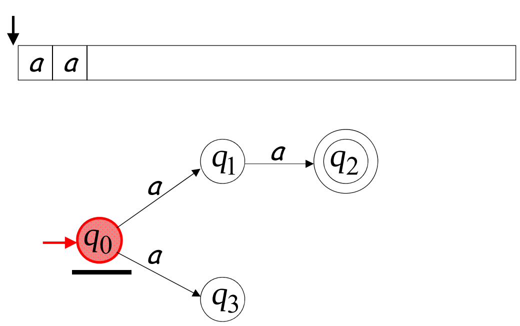
Two choices



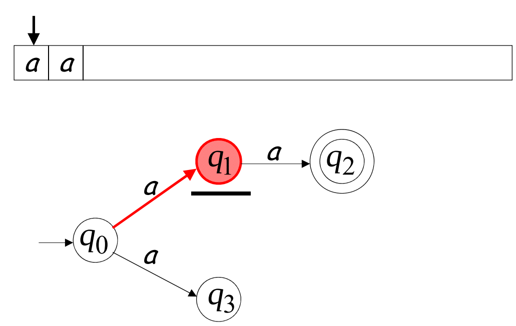
No transition

No transition

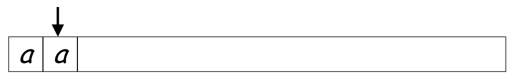
First Choice

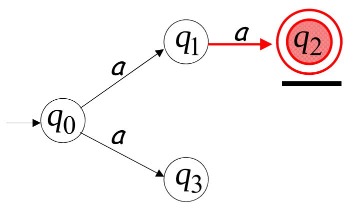


First Choice

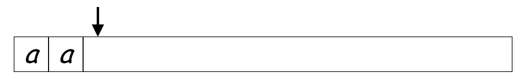


First Choice

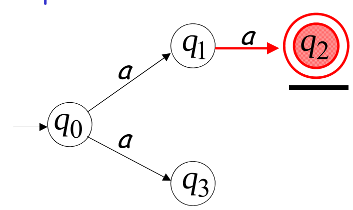




First Choice

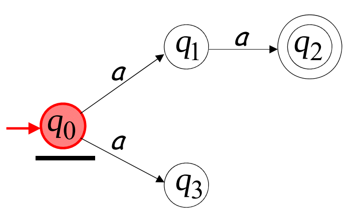


All input is consumed

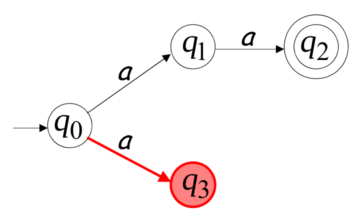


"accept"

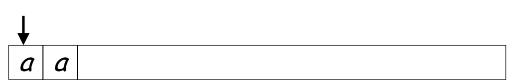
Second Choice

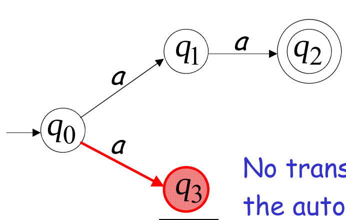


Second Choice



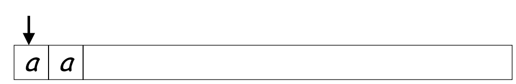
Second Choice



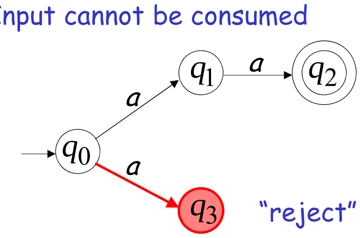


No transition:
the automaton hangs

Second Choice



Input cannot be consumed



"reject"

An NFA accepts a string:
when there is a computation of the NFA
that accepts the string

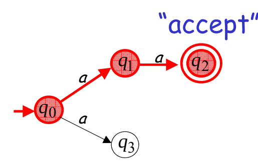
AND

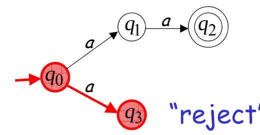
all the input is consumed and the automaton
is in a final state

13

Example

aa is accepted by the NFA:

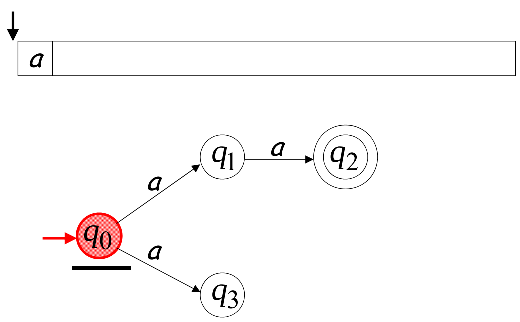




because this
computation
accepts aa

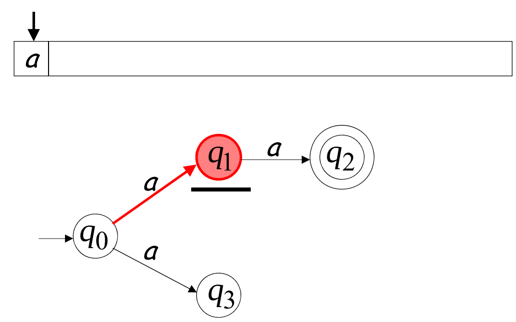
14

Rejection example



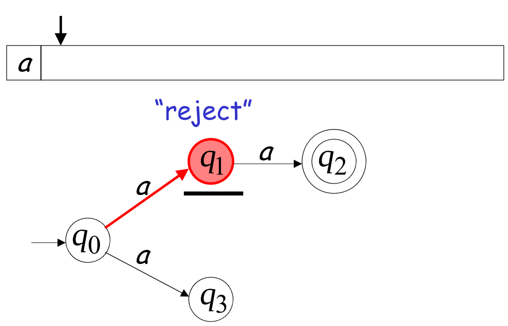
15

First Choice



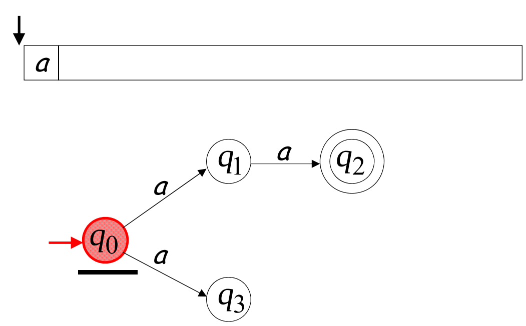
16

First Choice



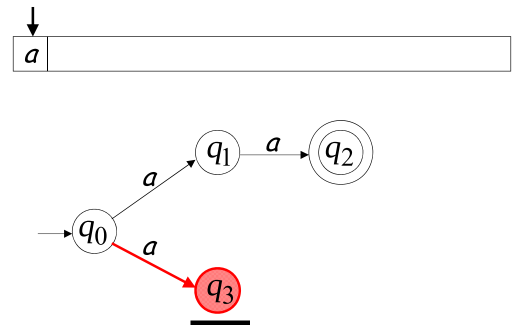
17

Second Choice



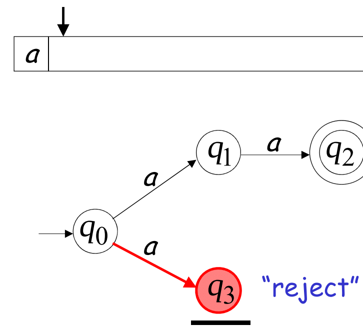
18

Second Choice



19

Second Choice



20

An NFA rejects a string:
when there is no computation of the NFA that accepts the string:

- All the input is consumed and the automaton is in a non final state

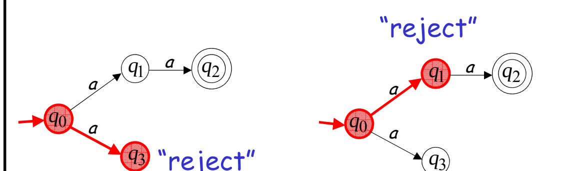
OR

- The input cannot be consumed

21

Example

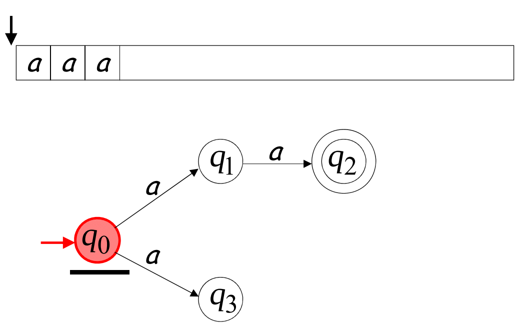
a is rejected by the NFA:



All possible computations lead to rejection

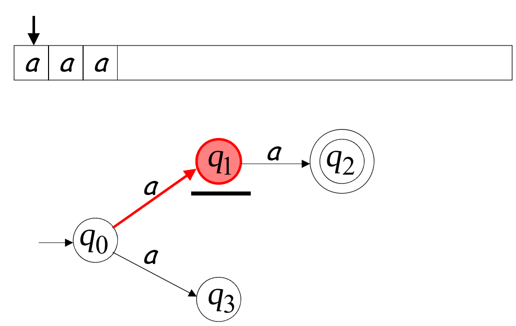
22

Rejection example



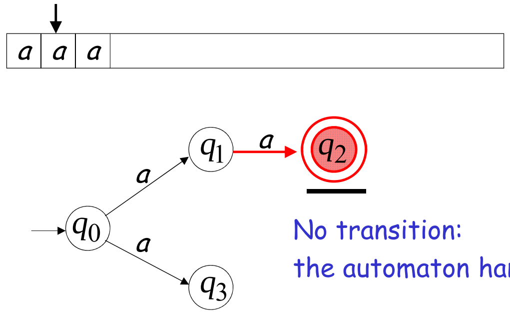
23

First Choice



24

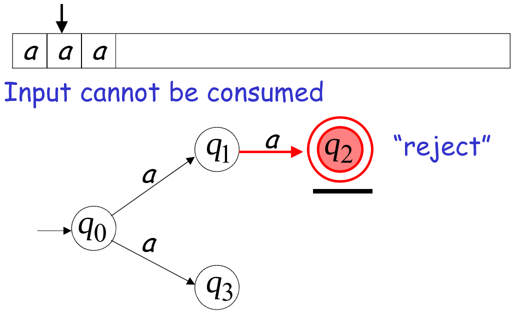
First Choice



No transition:
the automaton hangs

25

First Choice

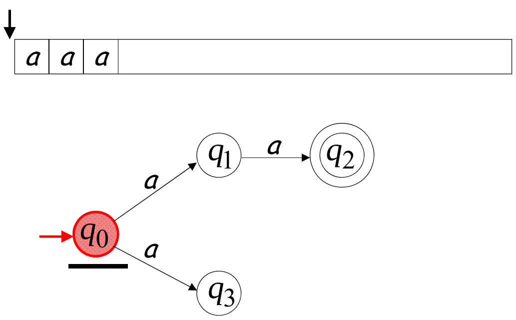


Input cannot be consumed

"reject"

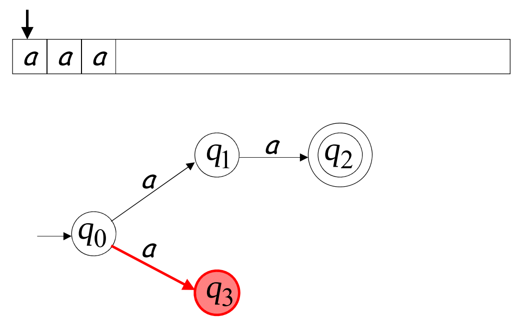
26

Second Choice



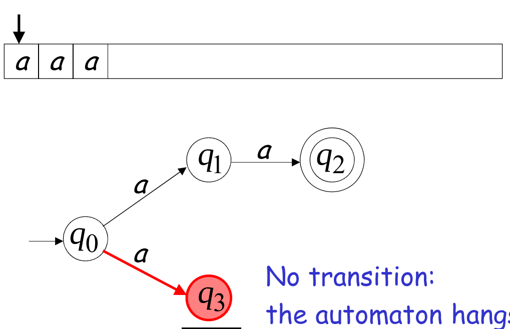
27

Second Choice



28

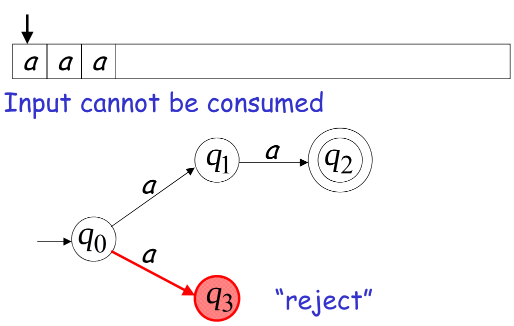
Second Choice



No transition:
the automaton hangs

29

Second Choice

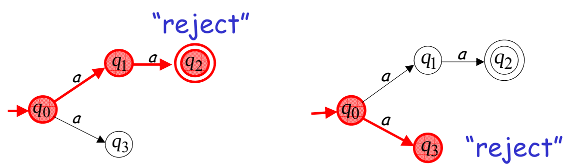


Input cannot be consumed

"reject"

30

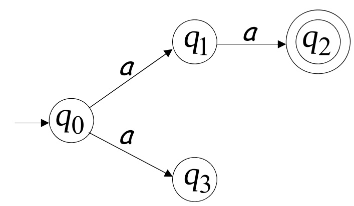
aaa is rejected by the NFA:



All possible computations lead to rejection

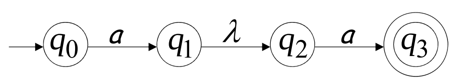
31

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

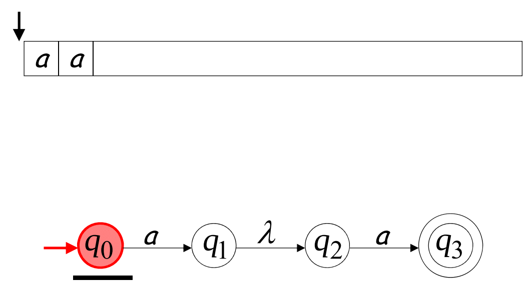


32

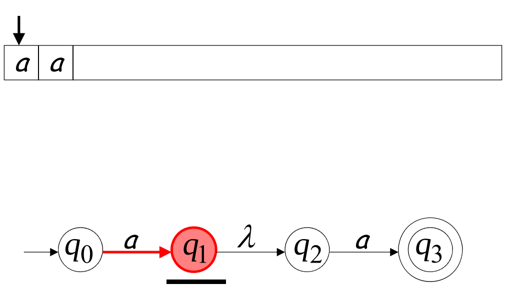
Lambda Transitions



33

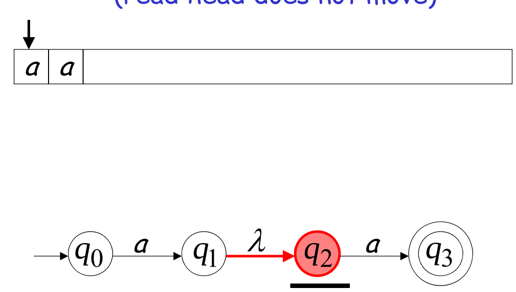


34

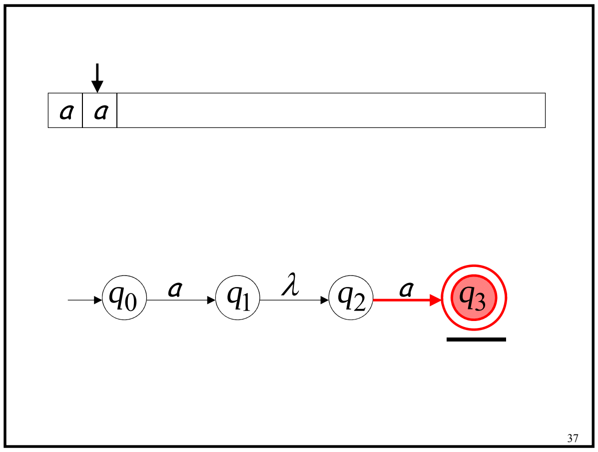


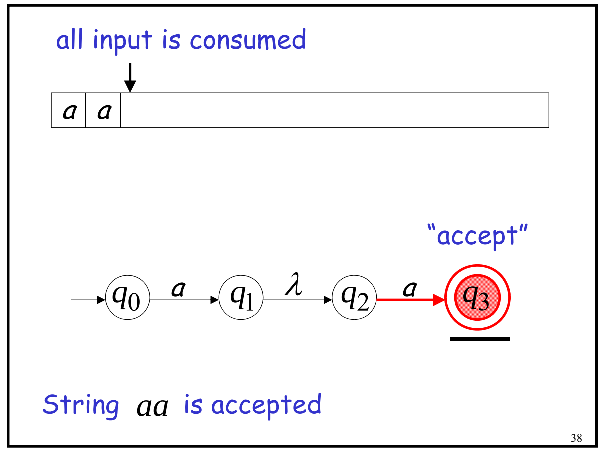
35

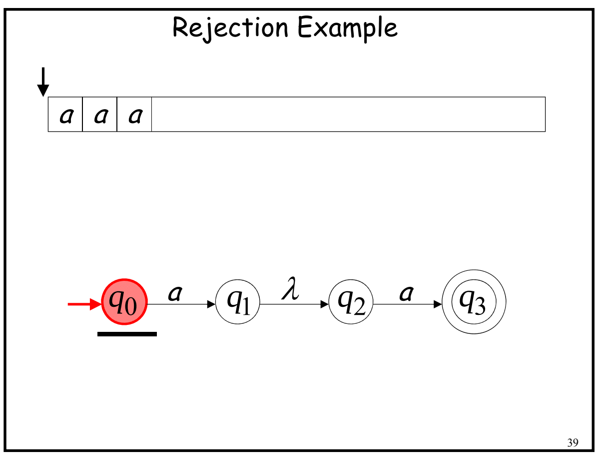
(read head does not move)

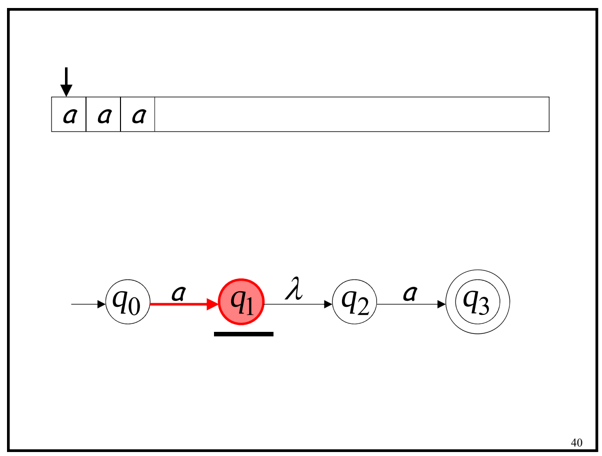


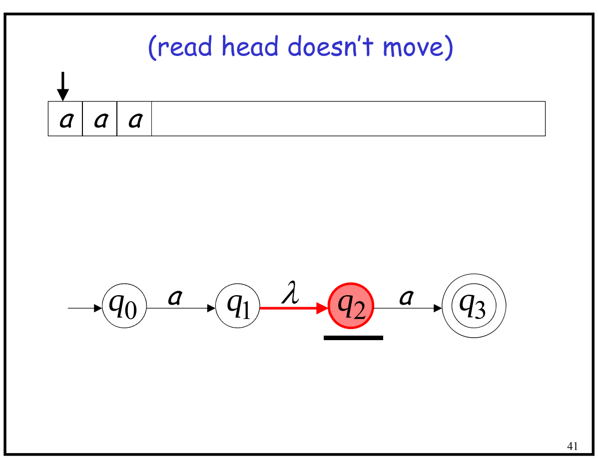
36

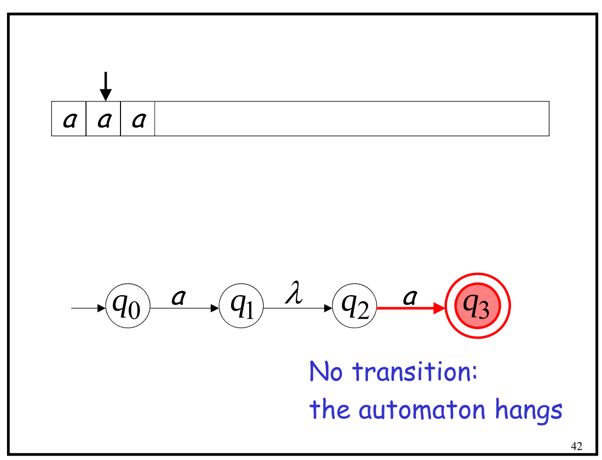

a a
q0 a q1 λ q2 a q3
37


all input is consumed
a a
"accept"
q0 a q1 λ q2 a q3
String aa is accepted
38

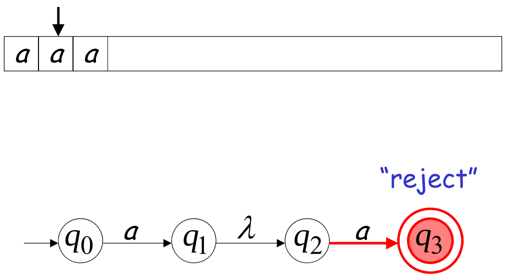

Rejection Example
a a a
q0 a q1 λ q2 a q3
39


a a a
q0 a q1 λ q2 a q3
40


(read head doesn't move)
a a a
q0 a q1 λ q2 a q3
41


a a a
q0 a q1 λ q2 a q3
No transition:
the automaton hangs
42

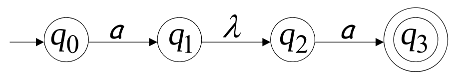
Input cannot be consumed



String aaa is rejected

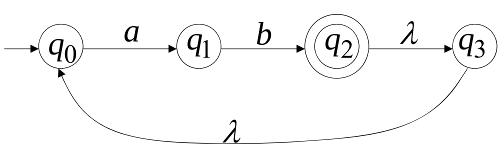
43

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

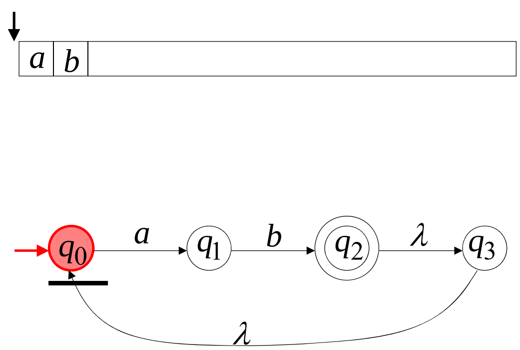


44

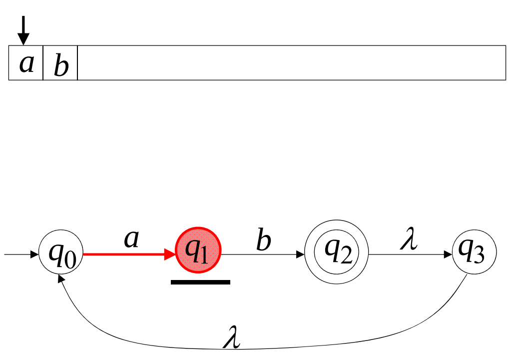
Another NFA Example



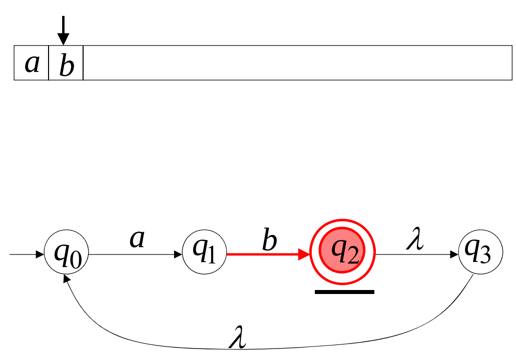
45



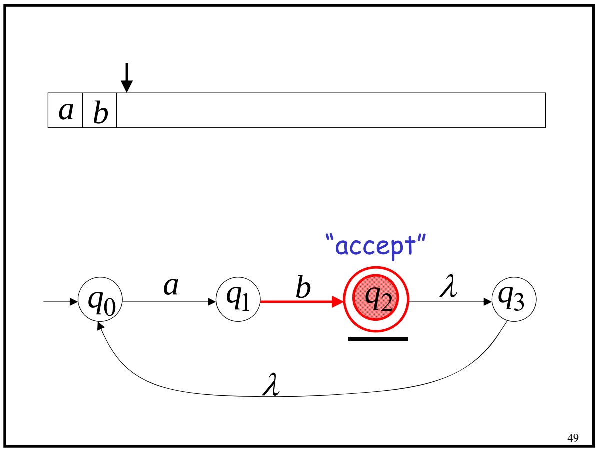
46



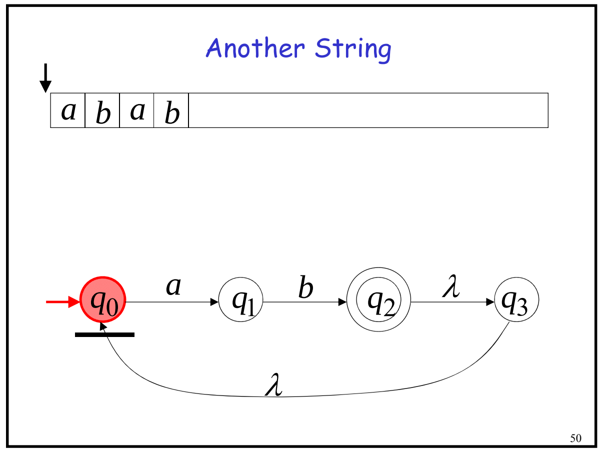
47



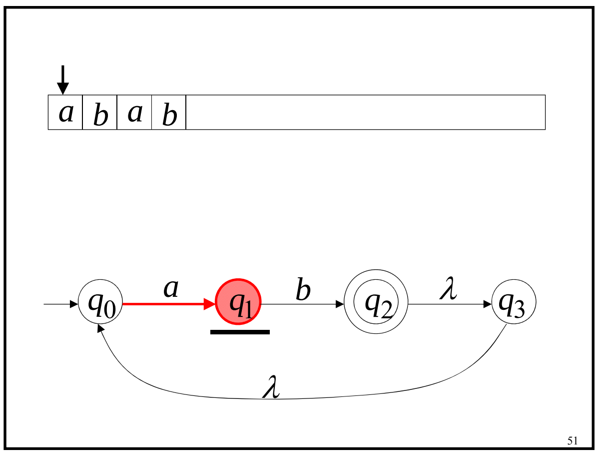
48



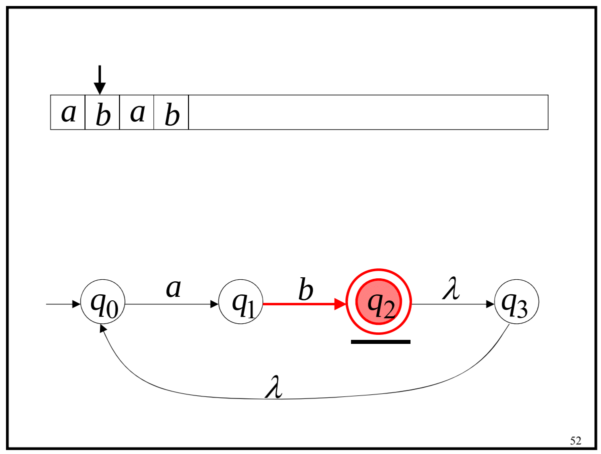
a b
"accept"
q_0 a q_1 b q_2 λ q_3
λ
49



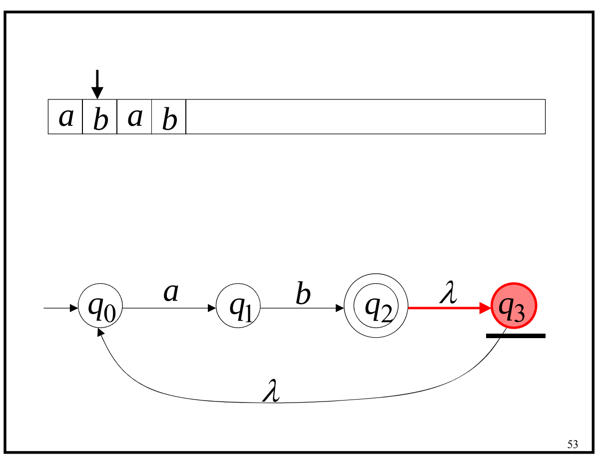
Another String
a b a b
q_0 a q_1 b q_2 λ q_3
λ
50



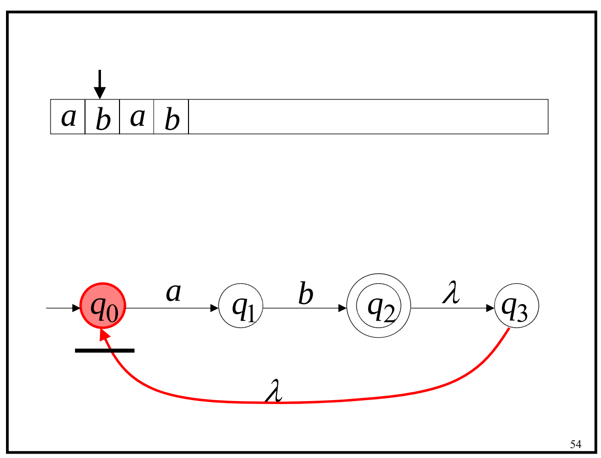
a b a b
q_0 a q_1 b q_2 λ q_3
λ
51



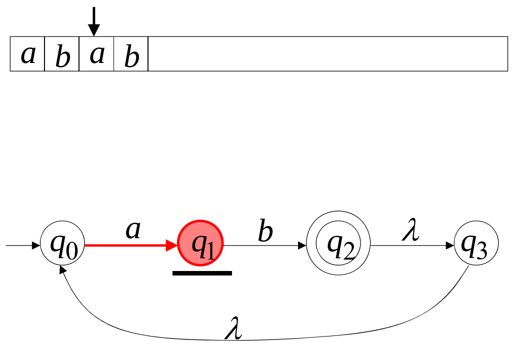
a b a b
q_0 a q_1 b q_2 λ q_3
λ
52

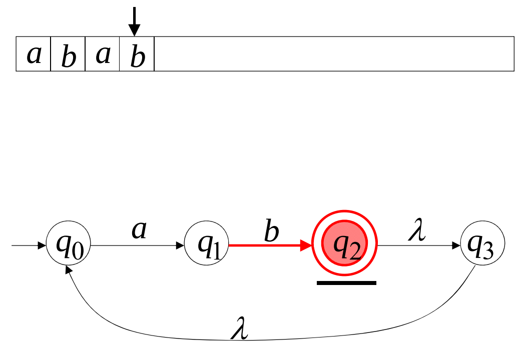


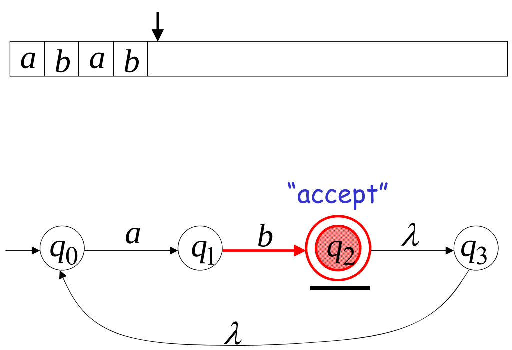
a b a b
q_0 a q_1 b q_2 λ q_3
λ
53



a b a b
q_0 a q_1 b q_2 λ q_3
λ
54

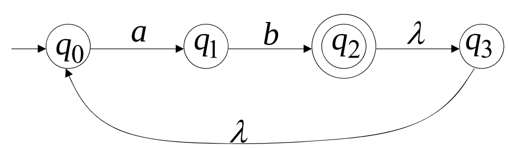




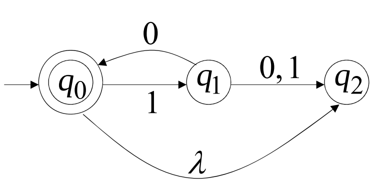


Language accepted

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

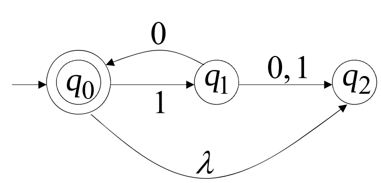


Another NFA Example



Language accepted

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



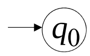
(redundant state)

Remarks:

- The λ symbol never appears on the input tape

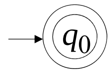
- Simple automata:

M_1



$L(M_1) = \{\}$

M_2

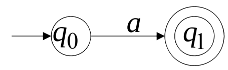


$L(M_2) = \{\lambda\}$

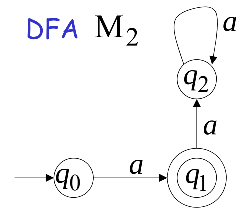
61

- NFAs are interesting because we can express languages easier than DFAs

NFA M_1



DFA M_2



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

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

62

Formal Definition of NFAs

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

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

Σ : Input aplhabet, i.e. $\{a, b\}$

δ : Transition function

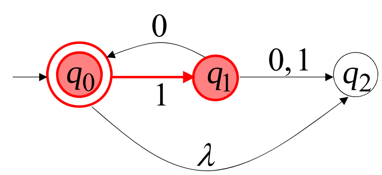
q_0 : Initial state

F : Final states

63

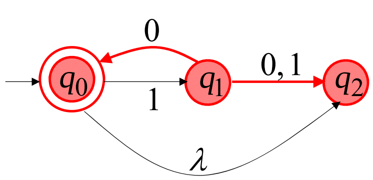
Transition Function δ

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



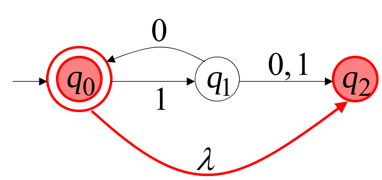
64

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



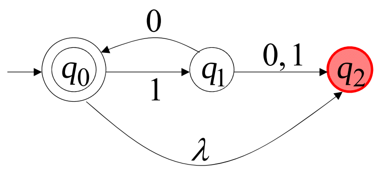
65

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



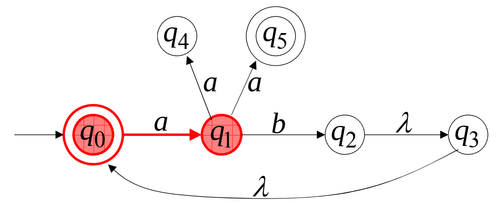
66

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

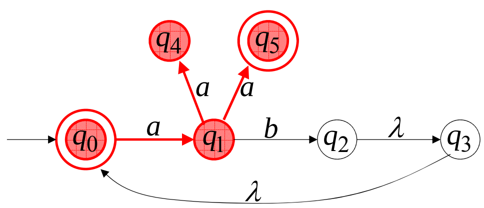


Extended Transition Function δ^*

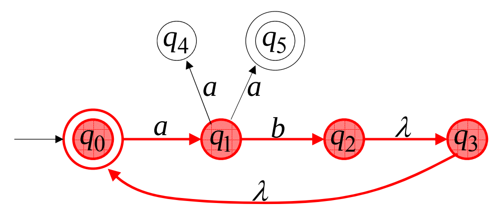
$$\delta^*(q_0, a) = \{q_1\}$$



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

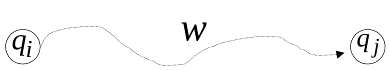


$$\delta^*(q_0, ab) = \{q_2, q_3, q_0\}$$

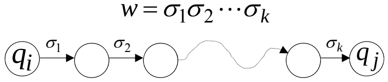


Formally

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

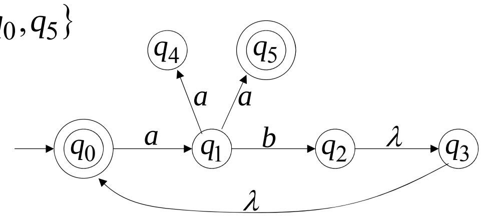


$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



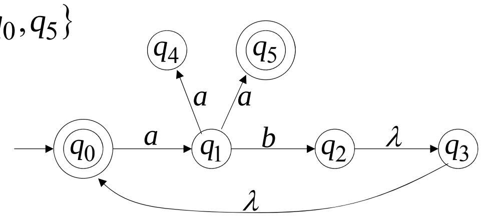
The Language of an NFA M

$F = \{q_0, q_5\}$



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

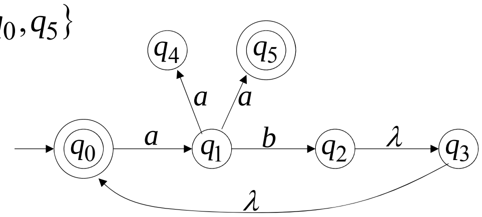
$F = \{q_0, q_5\}$



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

$\searrow \in F$

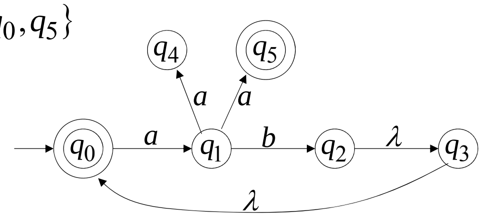
$F = \{q_0, q_5\}$



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

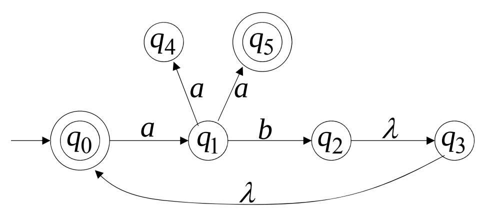
$\searrow \in F$

$F = \{q_0, q_5\}$



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

$\searrow \notin F$



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

Formally

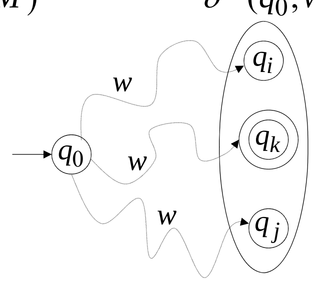
The language accepted by NFA M is:

$$L(M) = \{w_1, w_2, w_3, \ldots\}$$

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

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

$w \in L(M)$ $\qquad \delta^*(q_0, w)$



$q_k \in F$

NFAs accept the Regular Languages

79

Equivalence of Machines

Definition for Automata:

Machine M_1 is equivalent to machine M_2

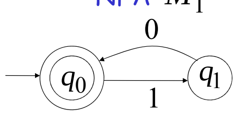
if $L(M_1) = L(M_2)$

80

Example of equivalent machines

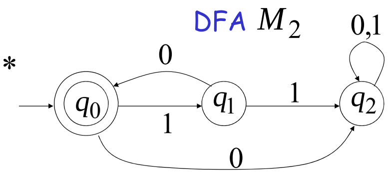
NFA M_1

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



DFA M_2

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



81

We will prove:

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

Languages accepted by DFAs

NFAs and DFAs have the same computation power

82

Step 1

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

Proof: Every DFA is trivially an NFA

⇓

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

83

Step 2

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

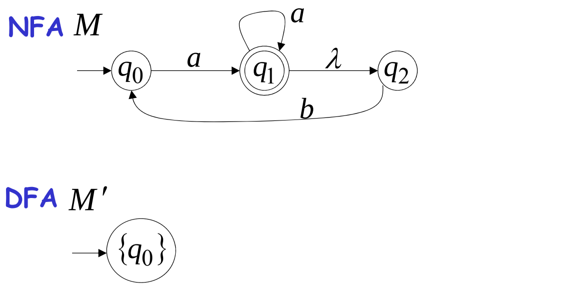
Proof: Any NFA can be converted to an equivalent DFA

⇓

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

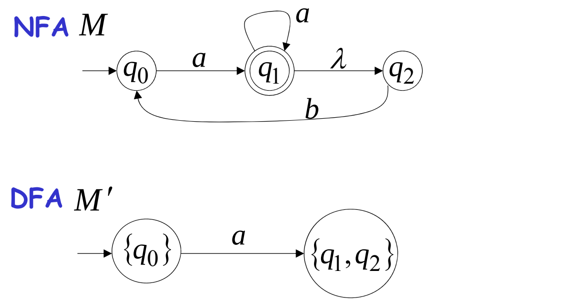
84

Convert NFA to DFA



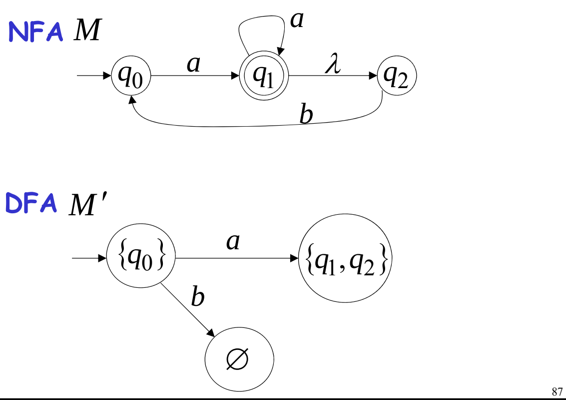
85

Convert NFA to DFA



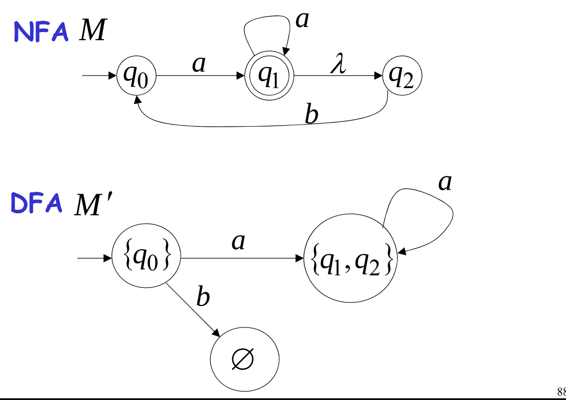
86

Convert NFA to DFA



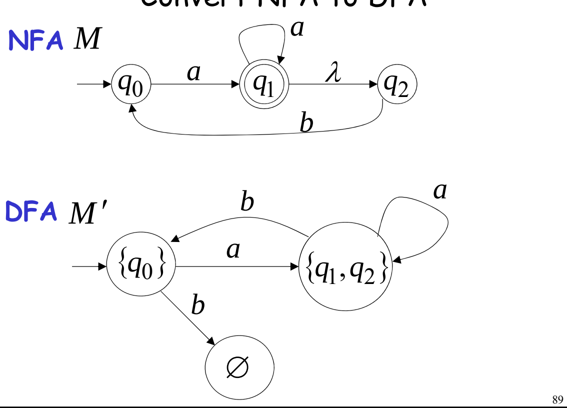
87

Convert NFA to DFA



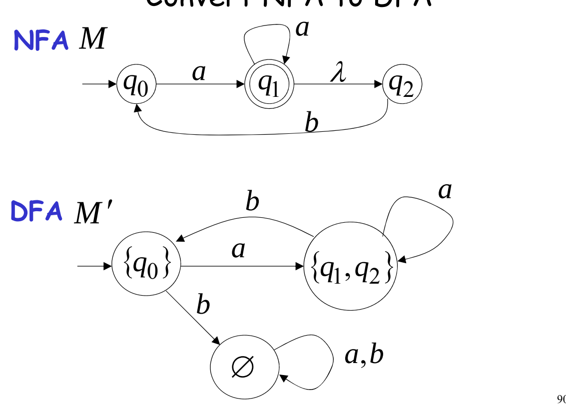
88

Convert NFA to DFA



89

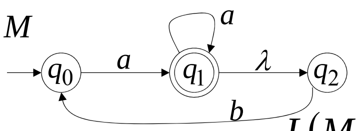
Convert NFA to DFA



90

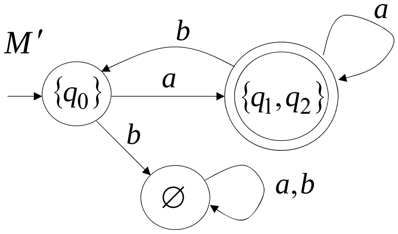
Convert NFA to DFA

NFA M



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

DFA M'



91

NFA to DFA: Remarks

We are given an NFA M

We want to convert it
to an equivalent DFA M'

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

92

If the NFA has states

$$q_0, q_1, q_2, \dots$$

the DFA has states in the powerset

$$\emptyset, \{q_0\}, \{q_1\}, \{q_1, q_2\}, \{q_3, q_4, q_7\}, \dots$$

93

Procedure NFA to DFA

1. Initial state of NFA: q_0

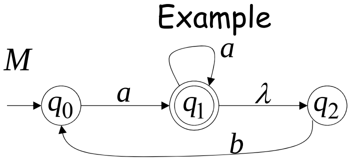
$\Downarrow$

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

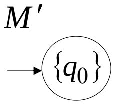
94

Example

NFA M



DFA M'



95

Procedure NFA to DFA

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

Compute in the NFA

$$\left.\begin{array}{l}\delta^*(q_i, a),\\ \delta^*(q_j, a),\\ \dots\end{array}\right\} = \{q'_i, q'_j, \dots, q'_m\}$$

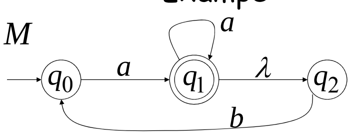
Add transition to DFA

$$\delta(\{q_i, q_j, \dots, q_m\},\ a) = \{q'_i, q'_j, \dots, q'_m\}$$

96

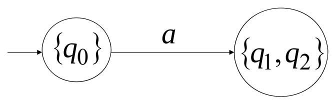
Exampe

NFA M



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

DFA M'



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

97

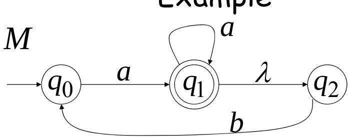
Procedure NFA to DFA

Repeat Step 2 for all letters in alphabet,
until
no more transitions can be added.

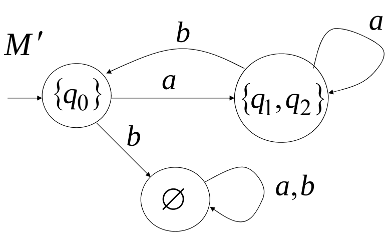
98

Example

NFA M



DFA M'



99

Procedure NFA to DFA

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

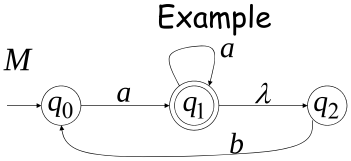
If some q_j is a final state in the NFA

Then, $\{q_i, q_j, \ldots, q_m\}$
is a final state in the DFA

100

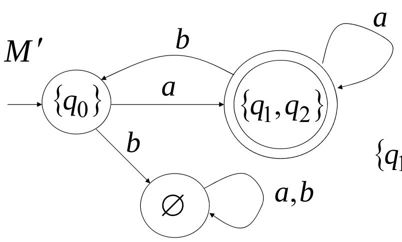
Example

NFA M



$q_1 \in F$

DFA M'



$\{q_1, q_2\} \in F'$

101

Theorem

Take NFA M

Apply procedure to obtain DFA M'

Then M and M' are equivalent :

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

102

Proof

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

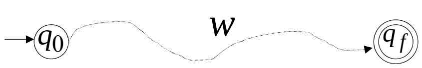
$$L(M) \subseteq L(M') \quad \text{AND} \quad L(M) \supseteq L(M')$$

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

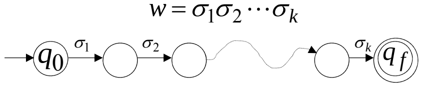
Take arbitrary: $w \in L(M)$

We will prove: $w \in L(M')$

$$w \in L(M)$$

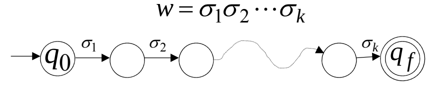
M : 

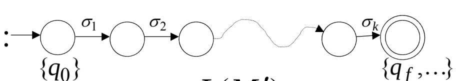
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

M : 

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

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

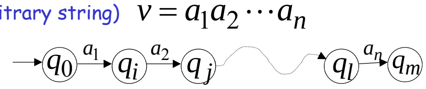
M : 

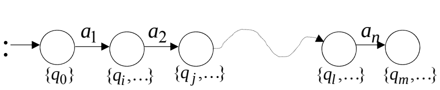
M' : 

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

More generally, we will show that if in M:

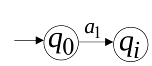
(arbitrary string) $v = a_1 a_2 \cdots a_n$

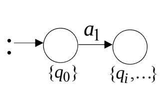
M : 

M' : 

Proof by induction on $|v|$

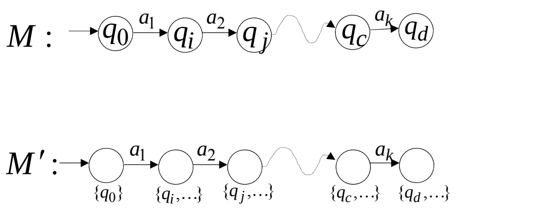
Induction Basis: $v = a_1$

M : 

M' : 

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

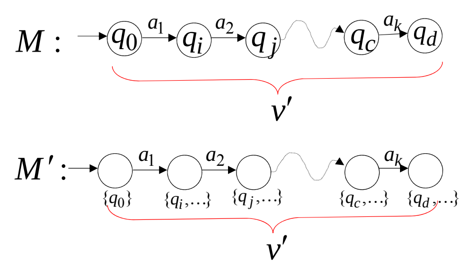
$$v = a_1 a_2 \cdots a_k$$



109

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

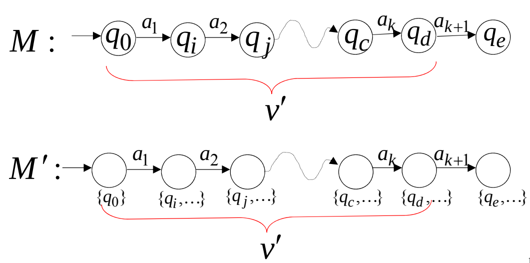
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$



110

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

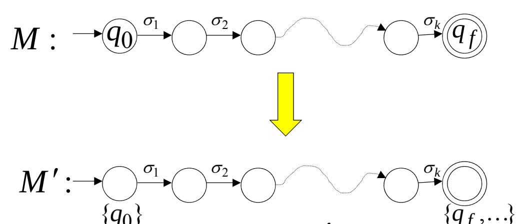
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$



111

Therefore if $w \in L(M)$

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



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

112

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

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

(proof is similar)

113