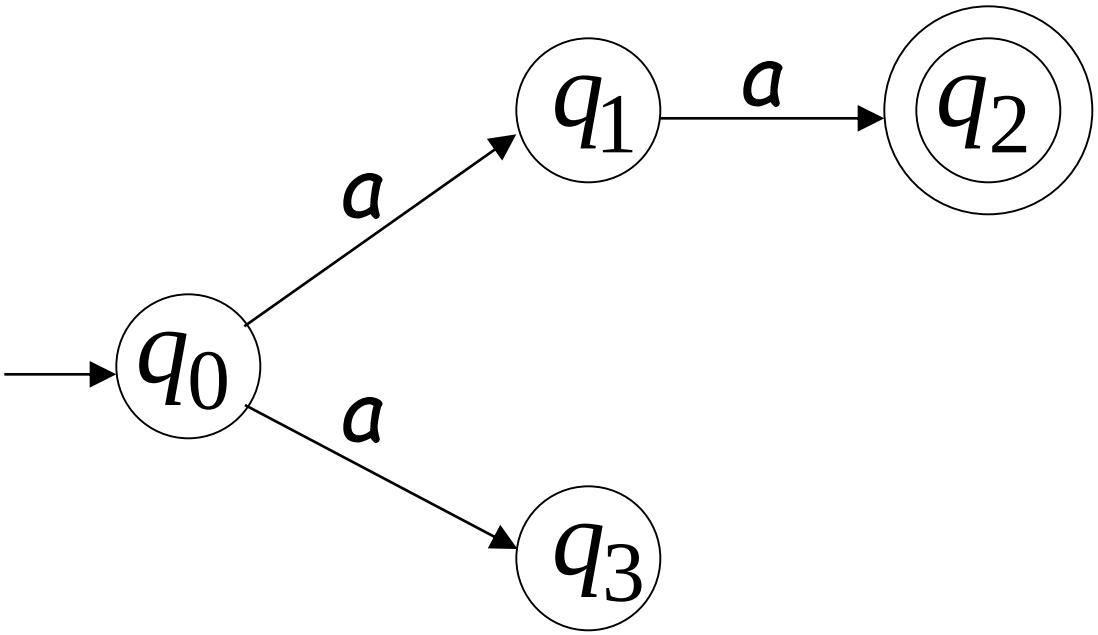
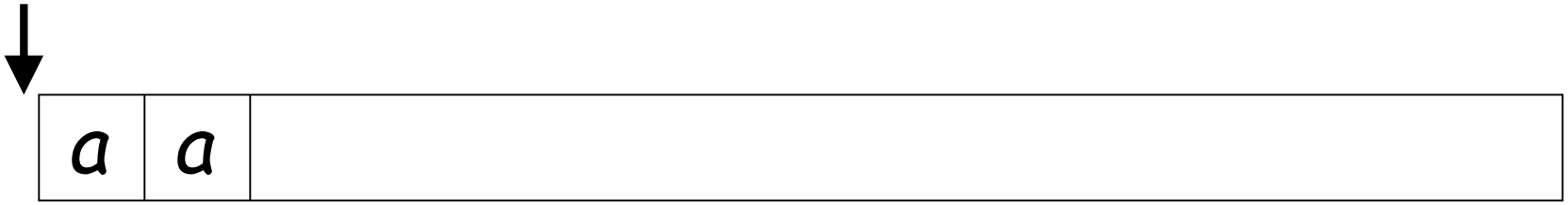
Non-Deterministic Finite Automata

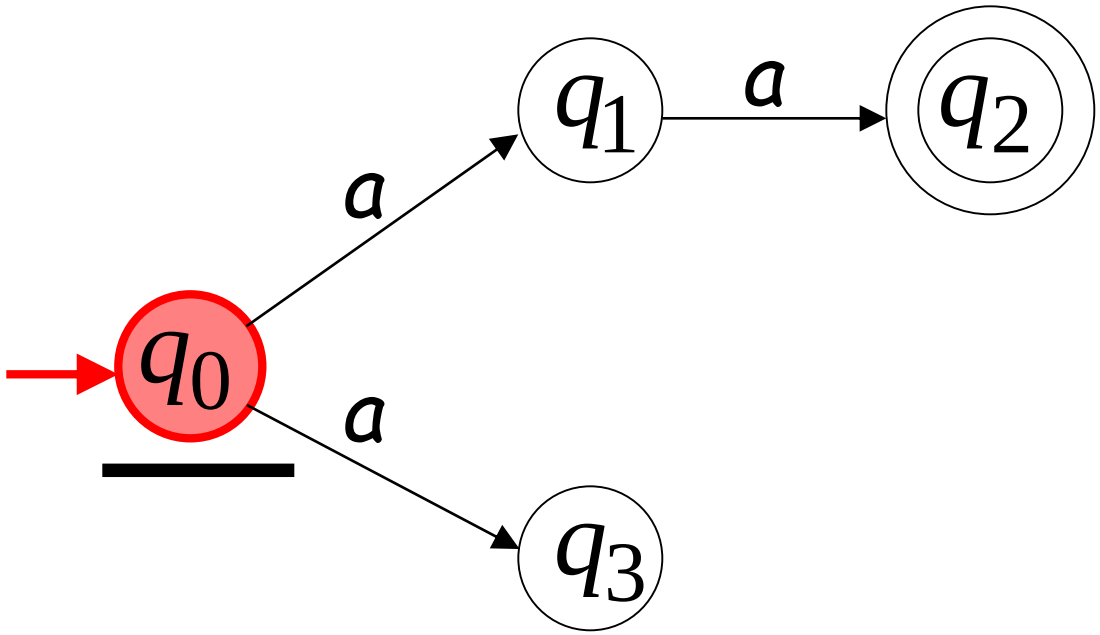
Nondeterministic Finite Automaton (NFA)

Alphabet = $\{a\}$

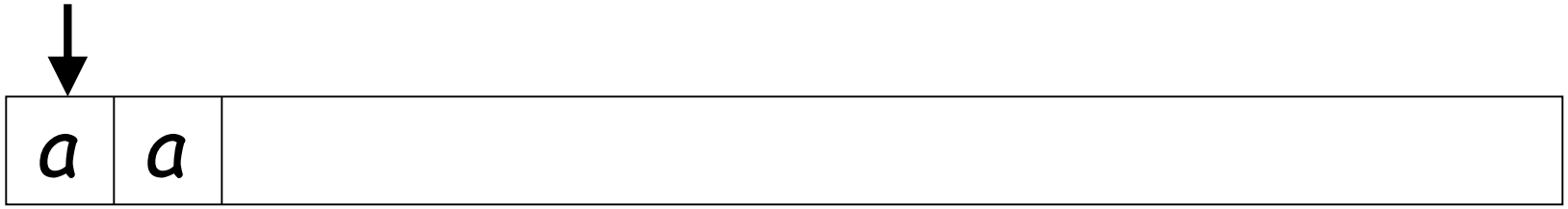


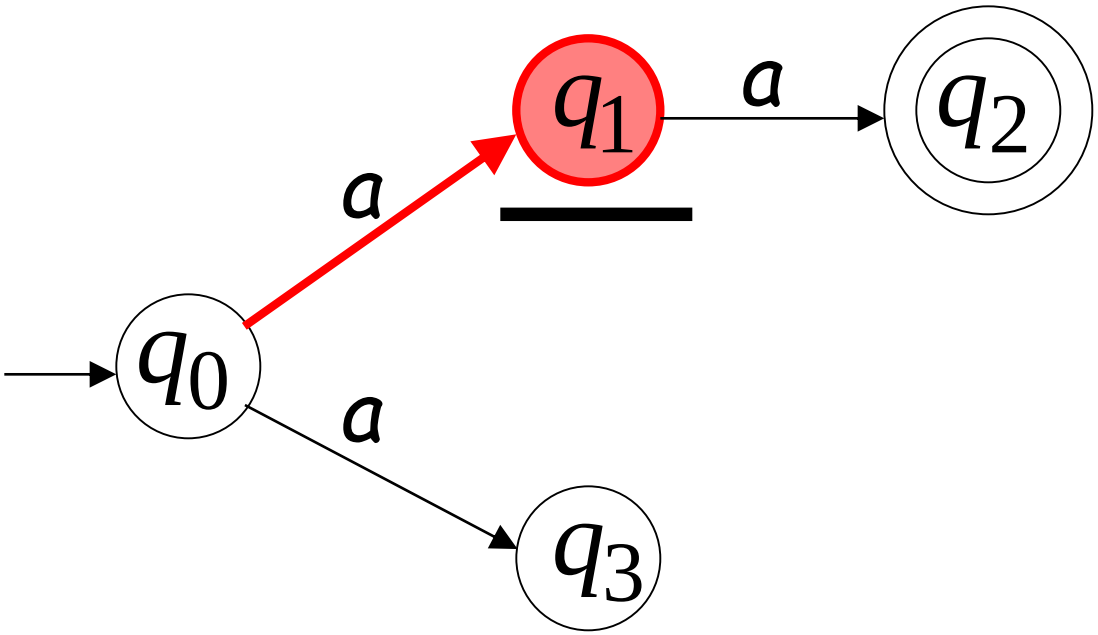
First Choice



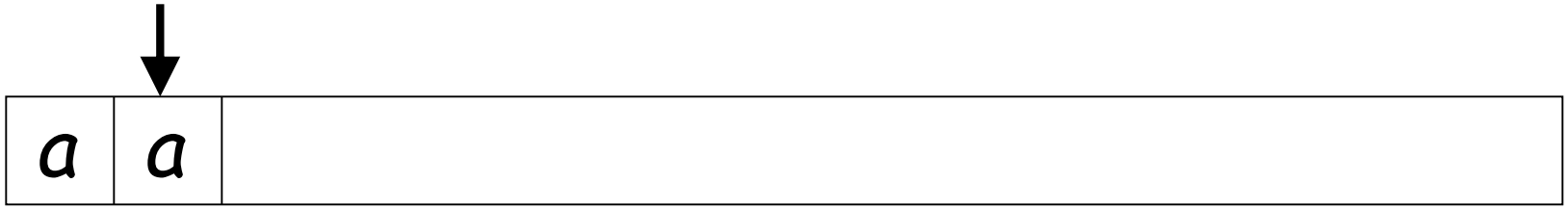


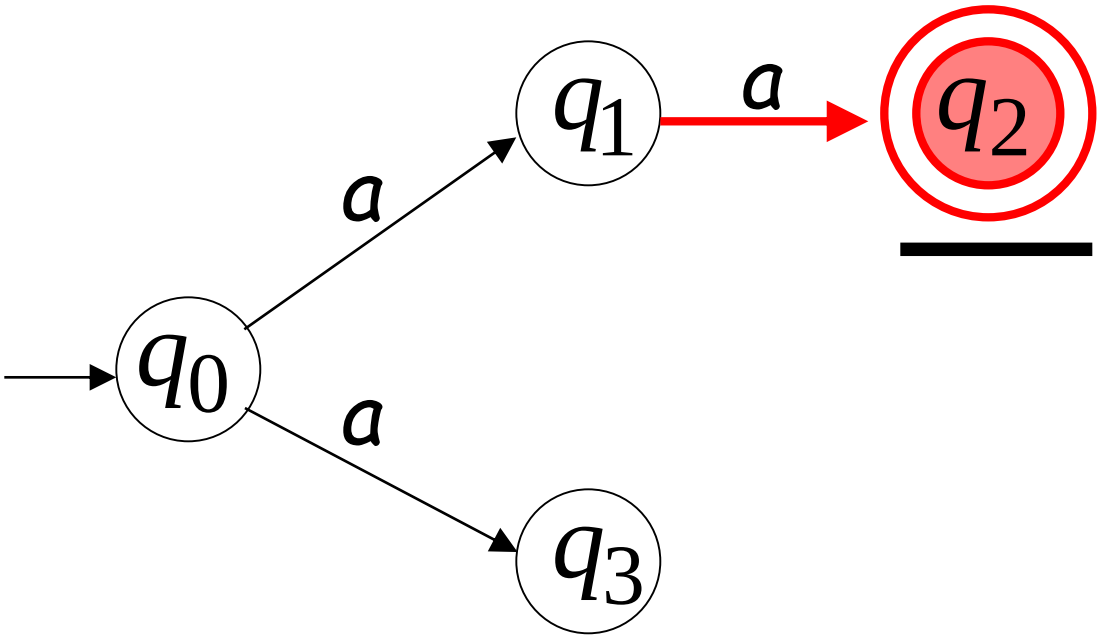
First Choice



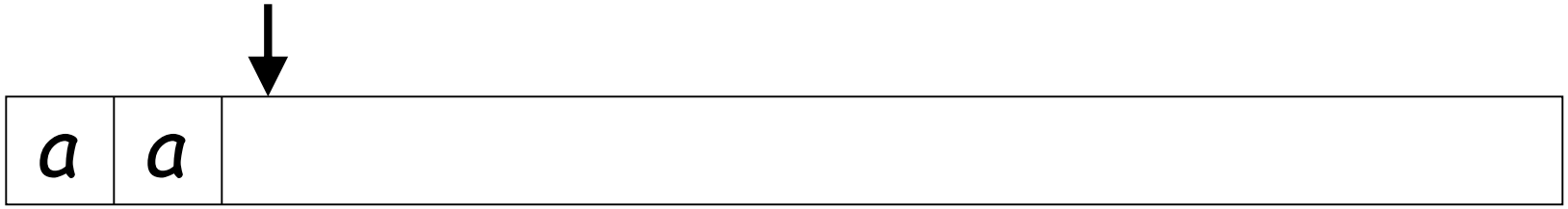


First Choice

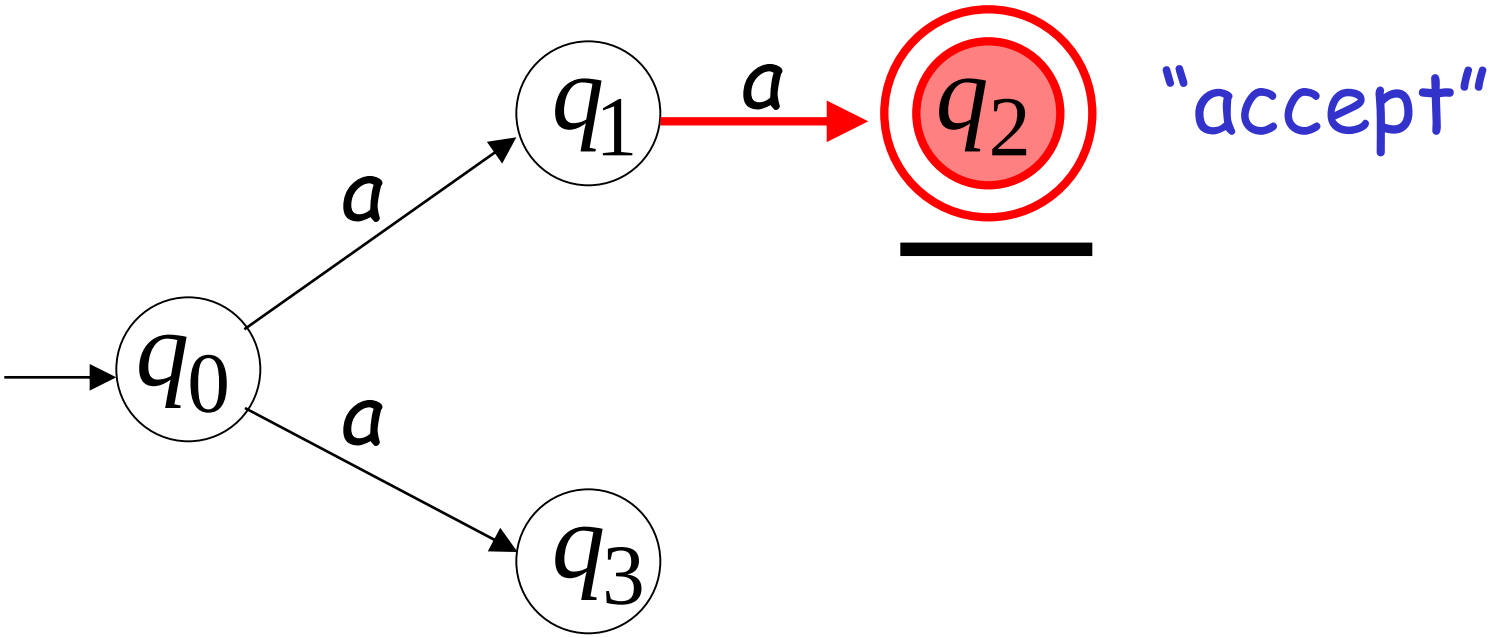




First Choice

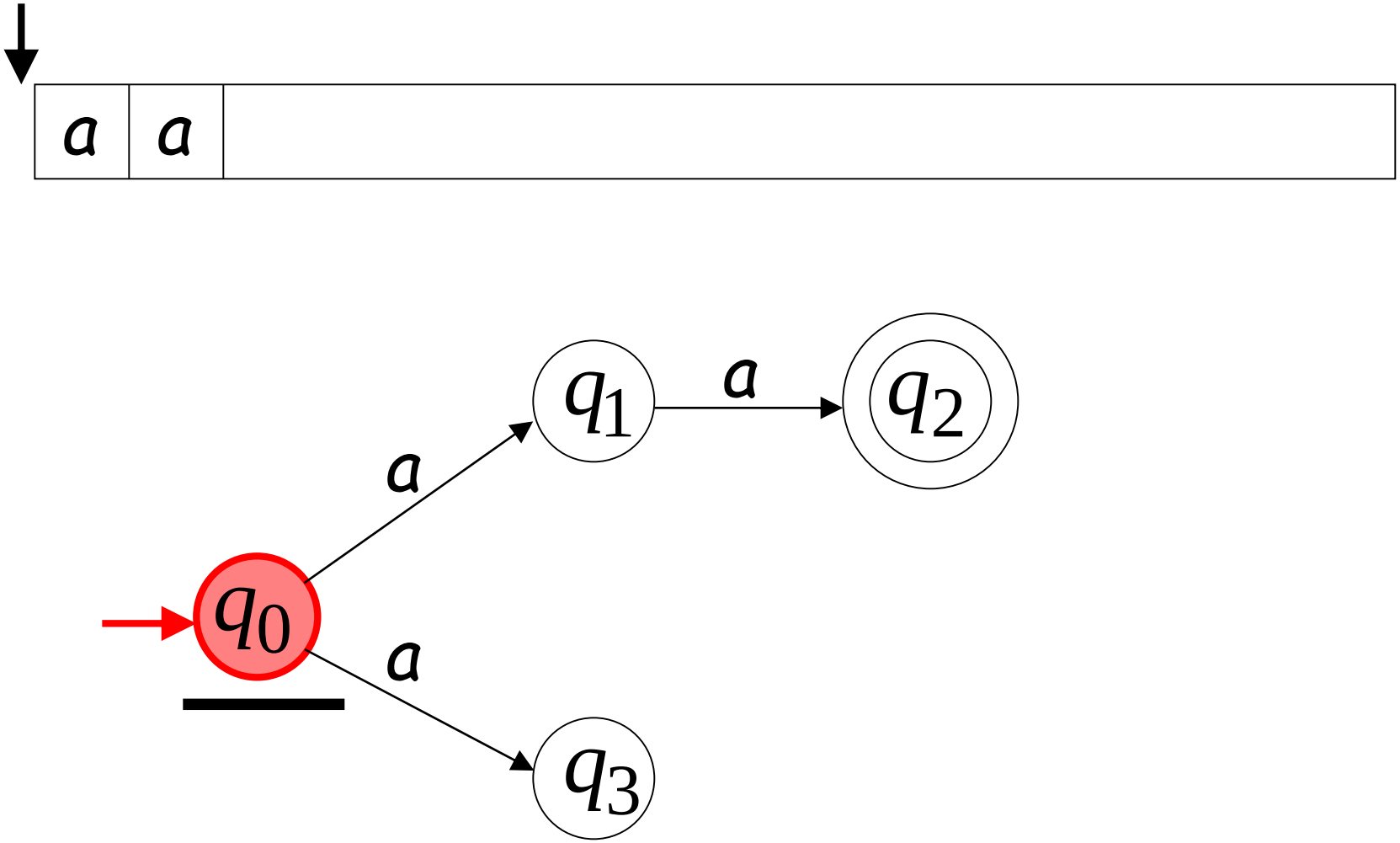


All input is consumed

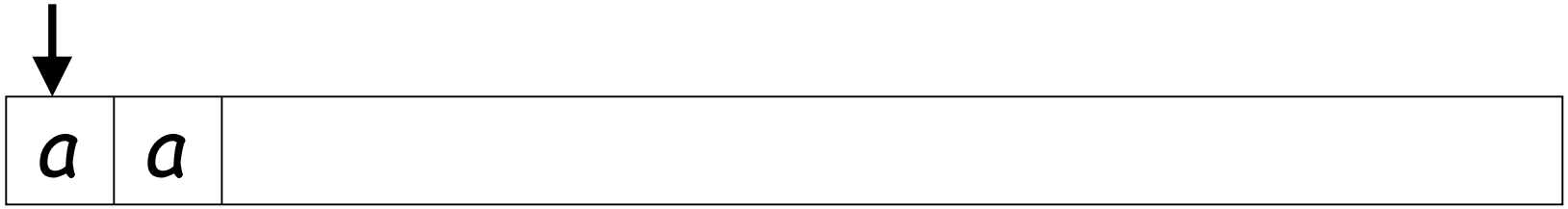


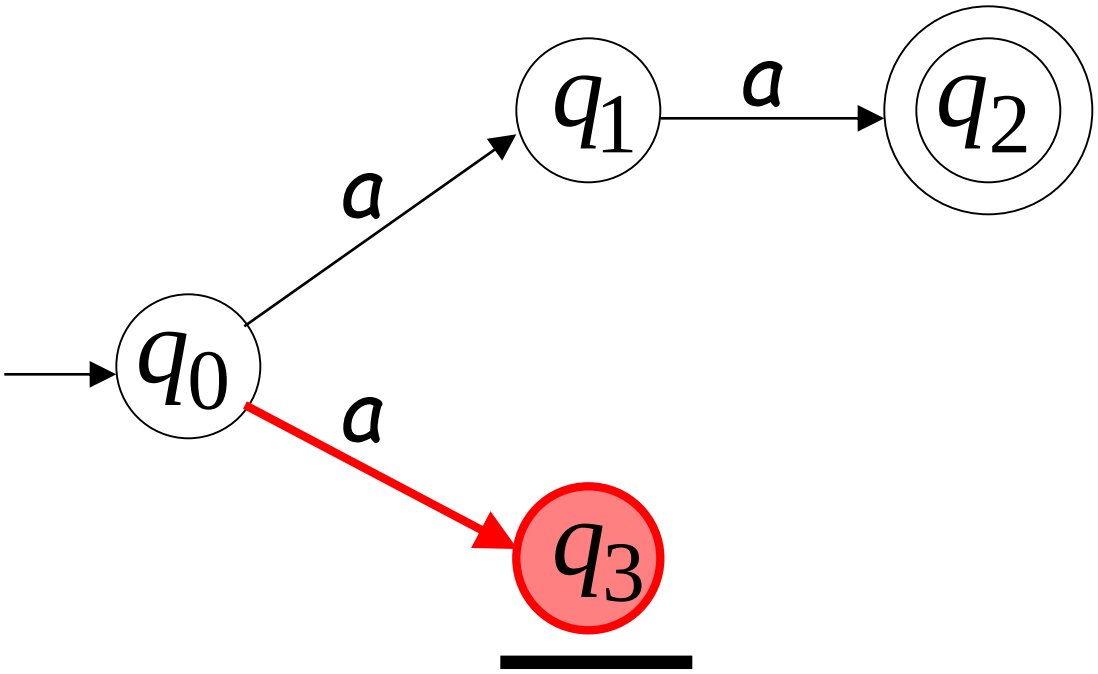
"accept"

Second Choice

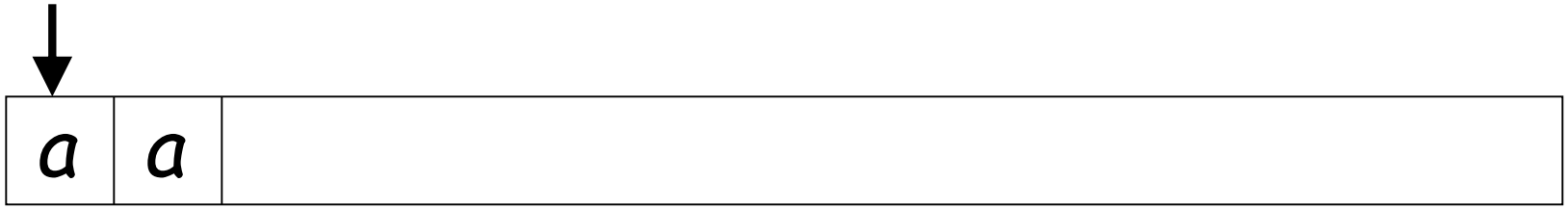


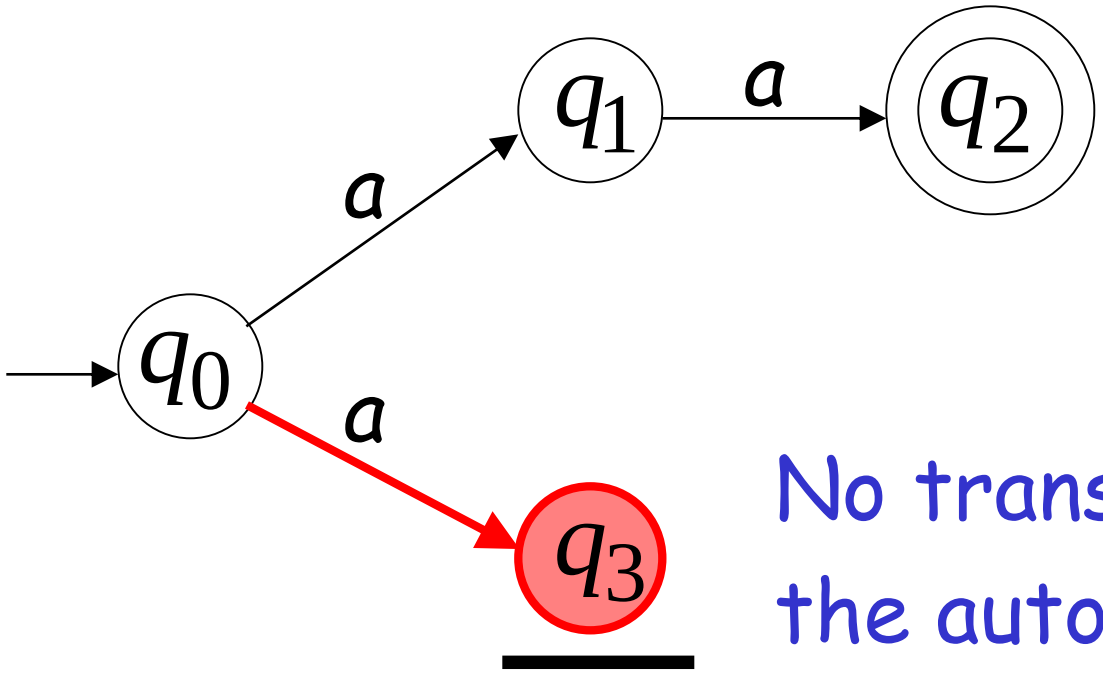
Second Choice





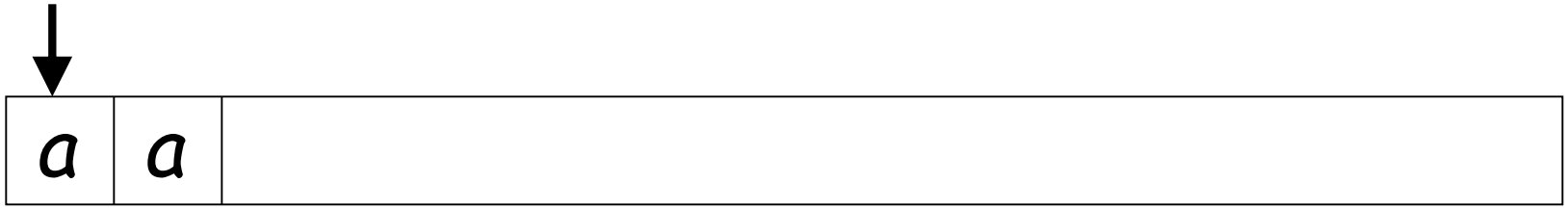
Second Choice



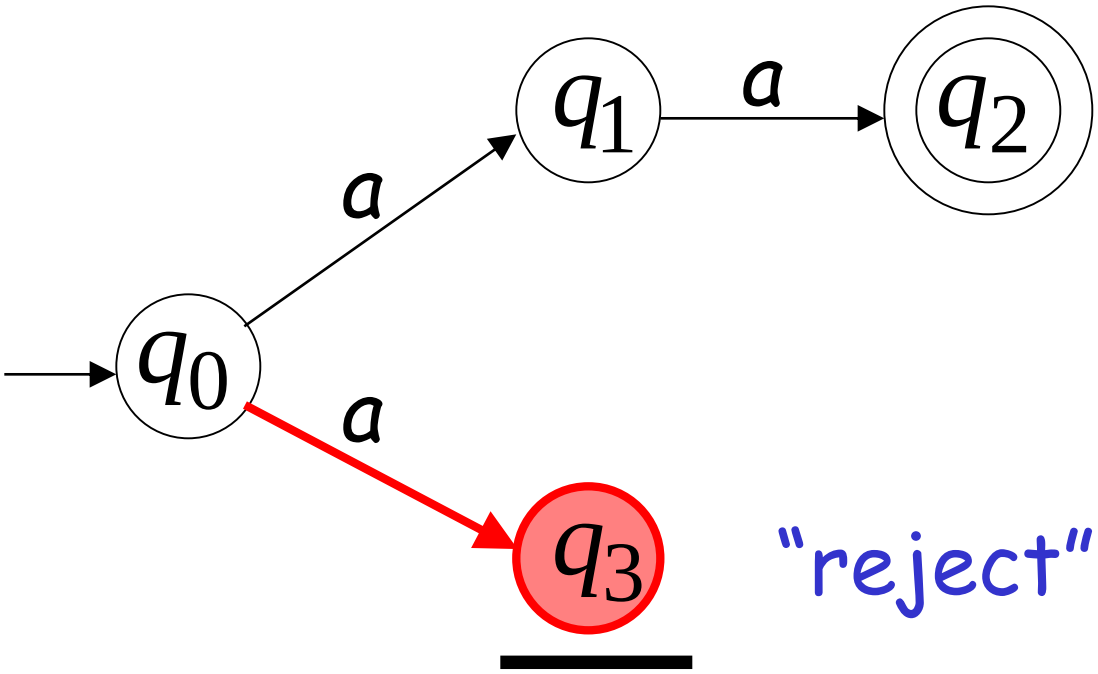


No transition:
the automaton hangs

Second Choice



Input cannot be consumed



"reject"

An NFA accepts a string:

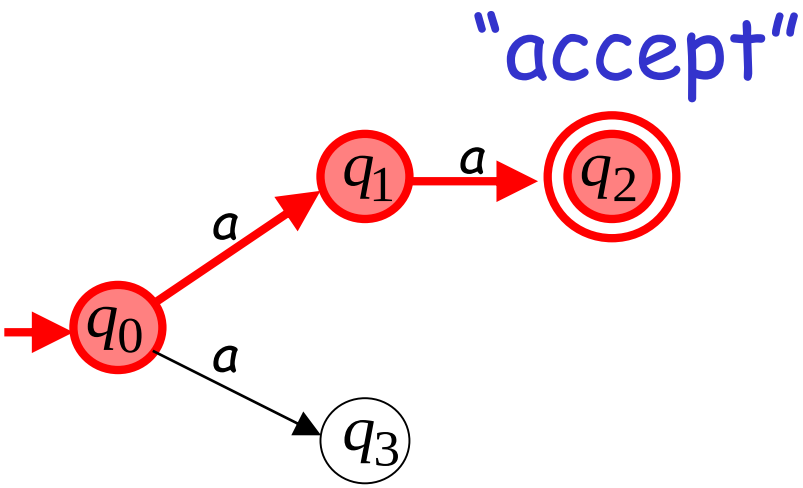
when there is a computation of the NFA that accepts the string

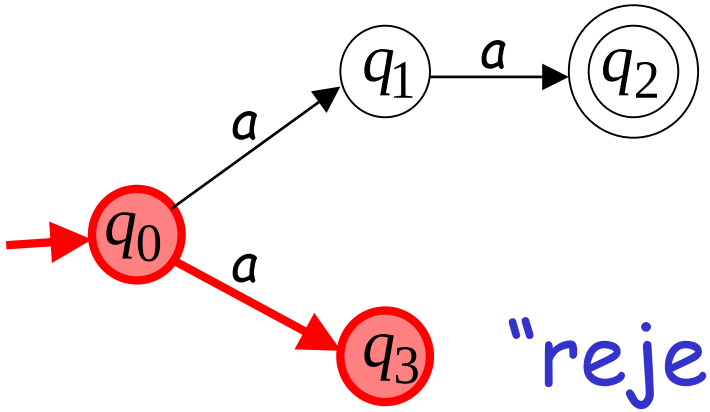
There is a computation:

all the input is consumed and the automaton is in an accepting state

Example

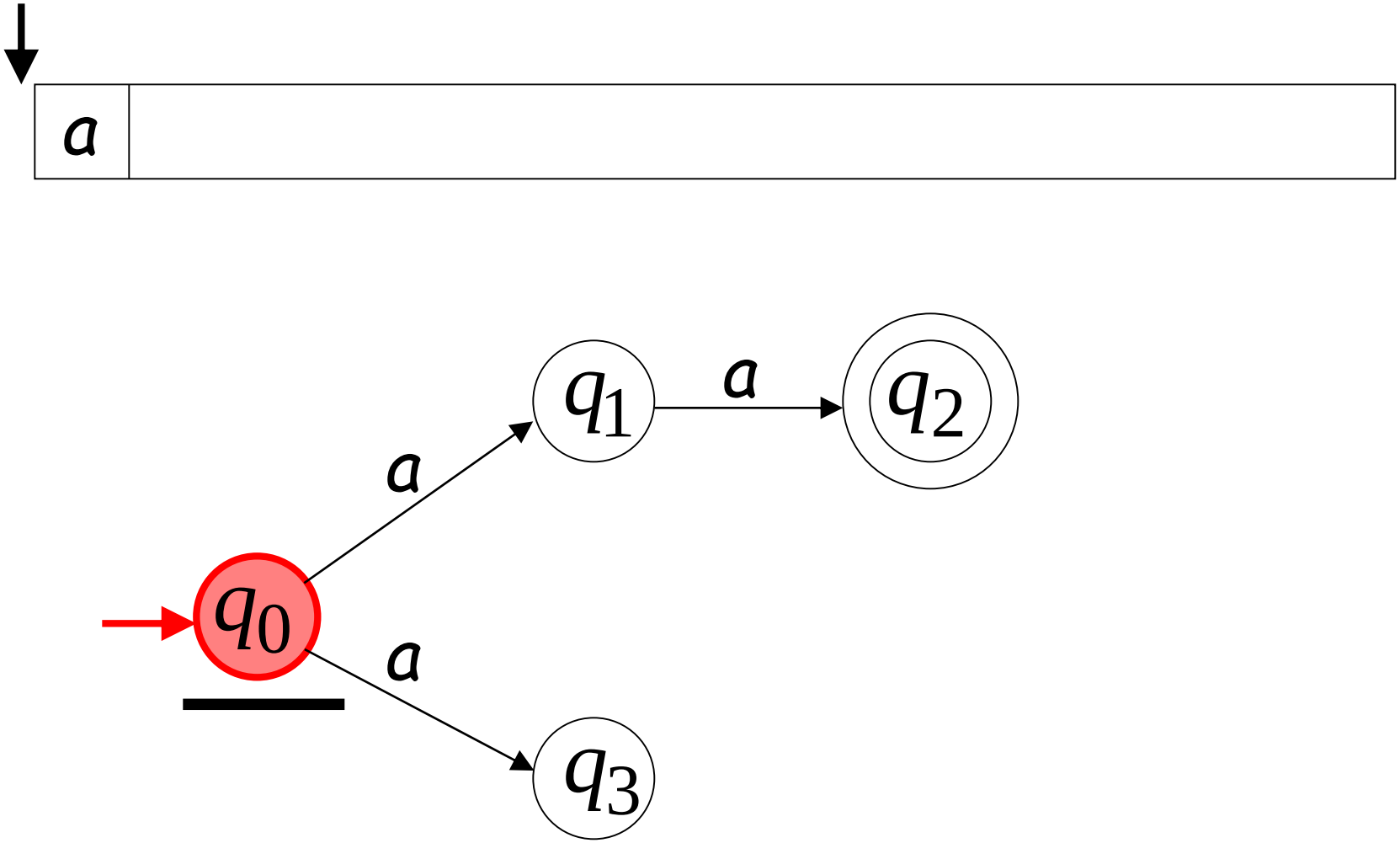
aa is accepted by the NFA:



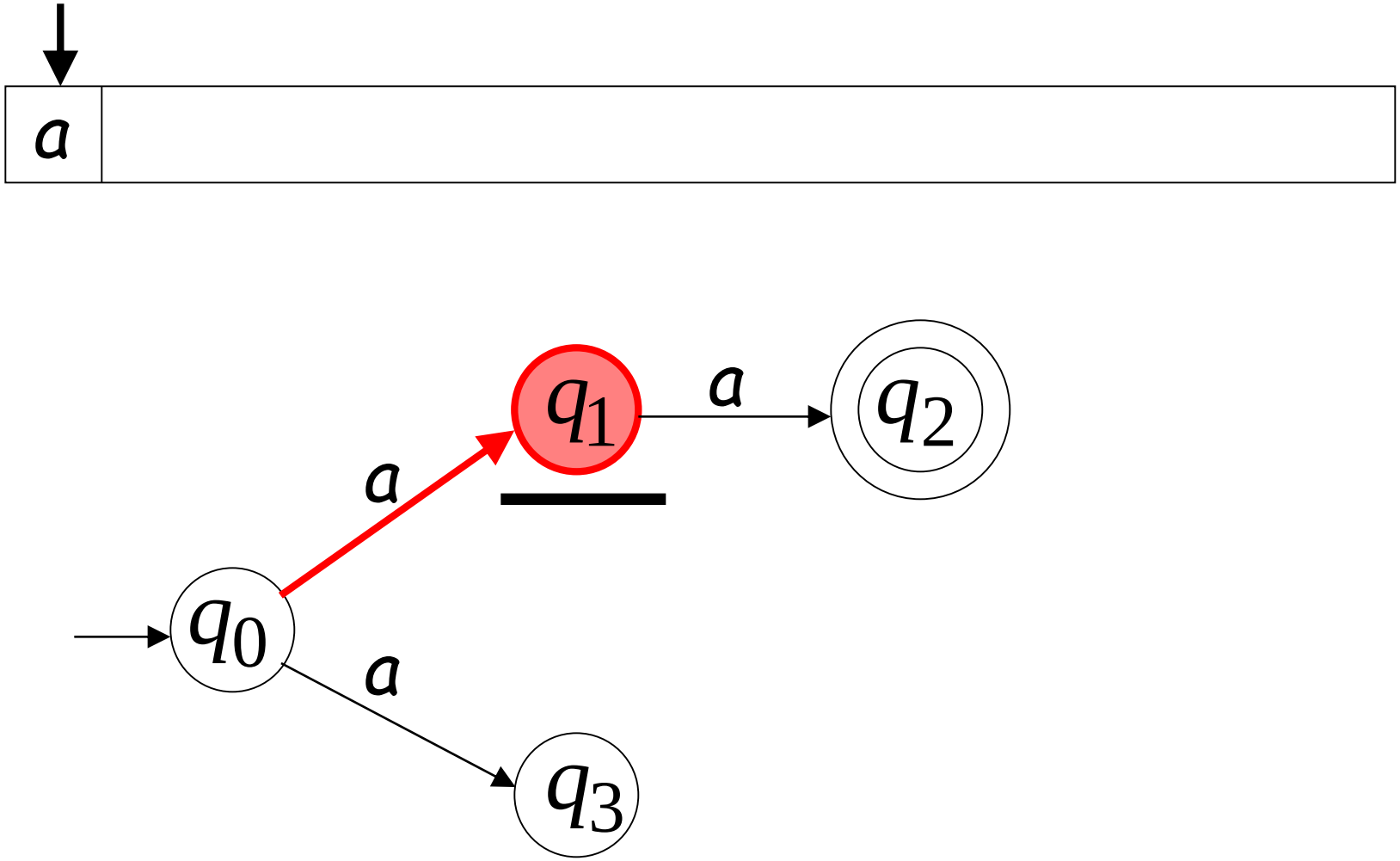


because this computation accepts aa

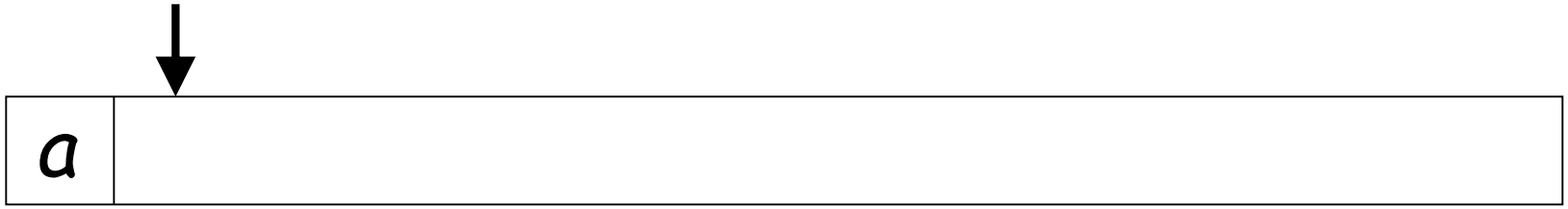
Rejection example



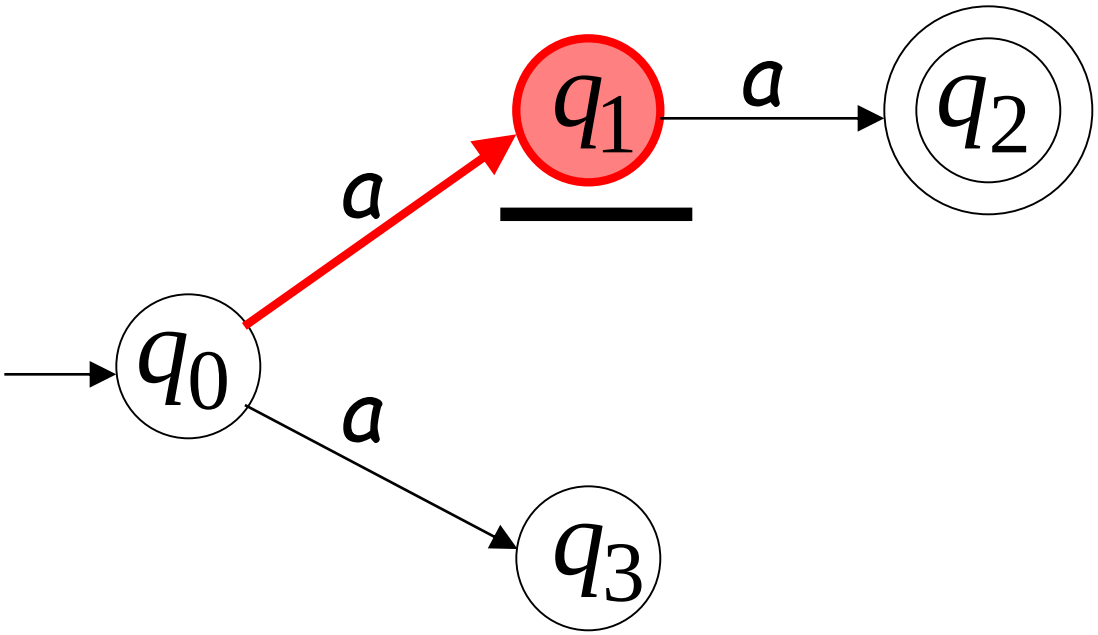
First Choice



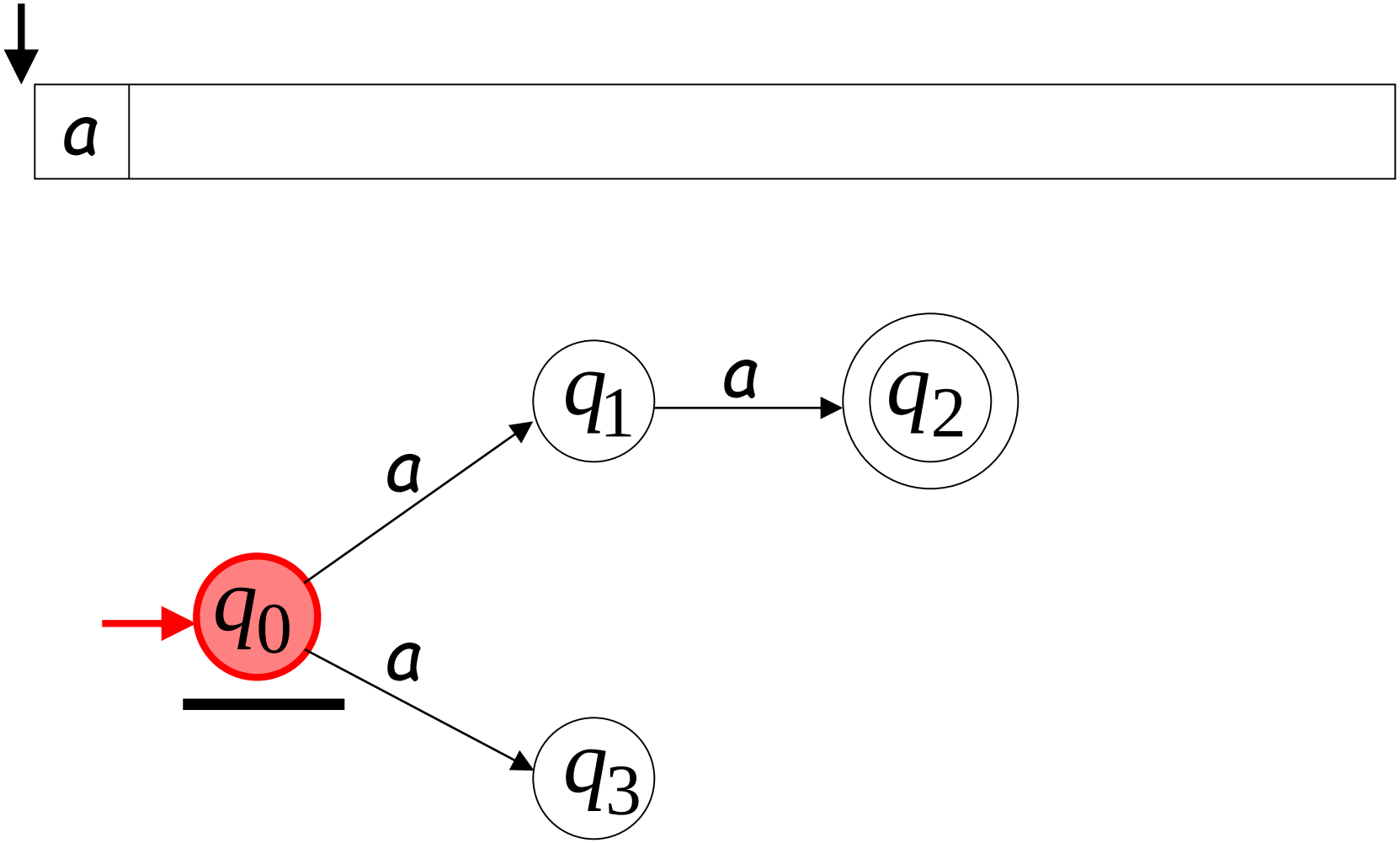
First Choice



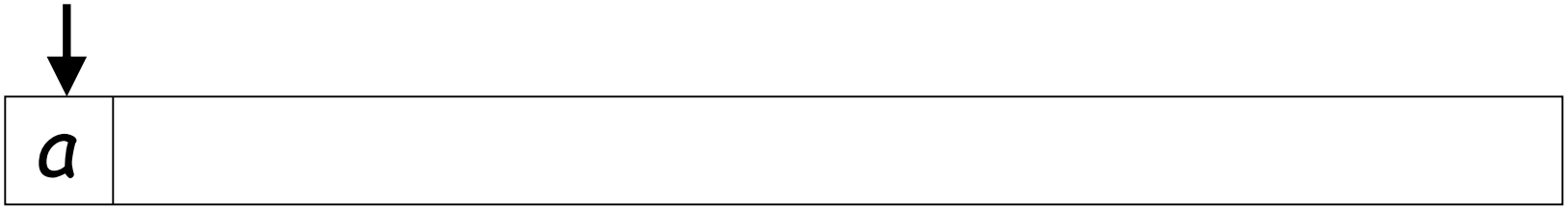
"reject"

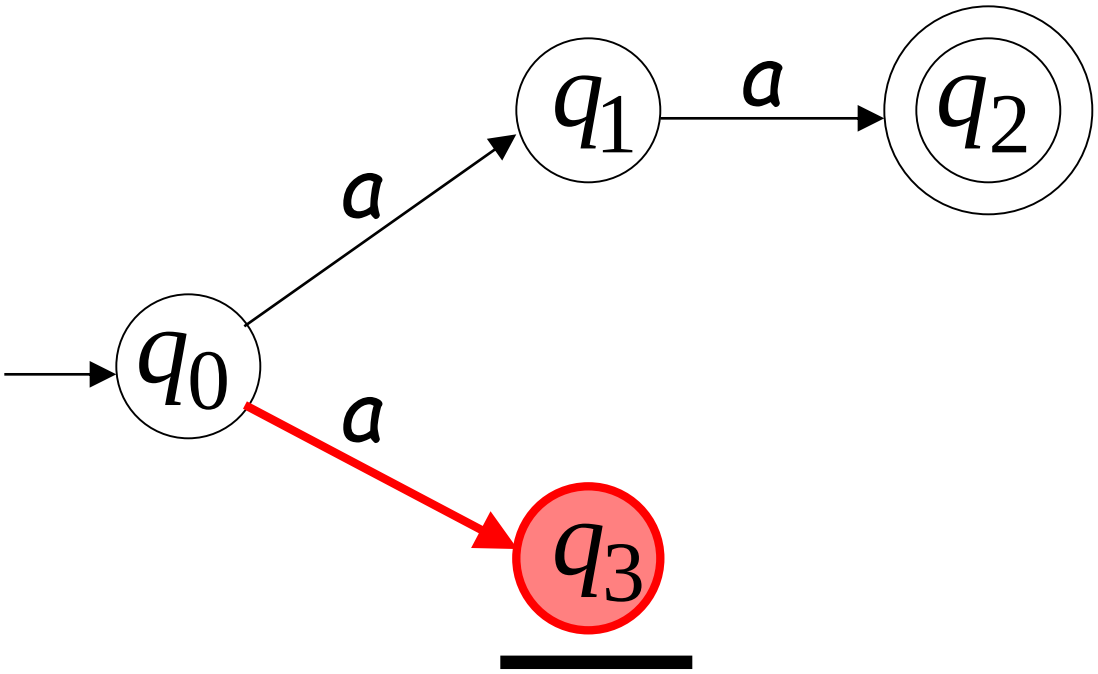


Second Choice

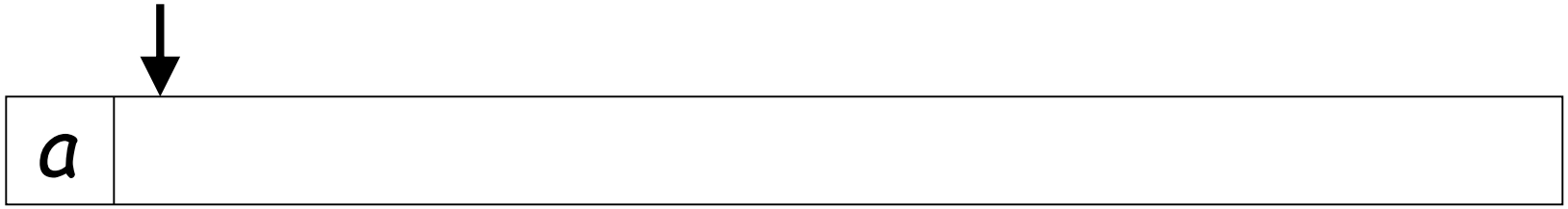


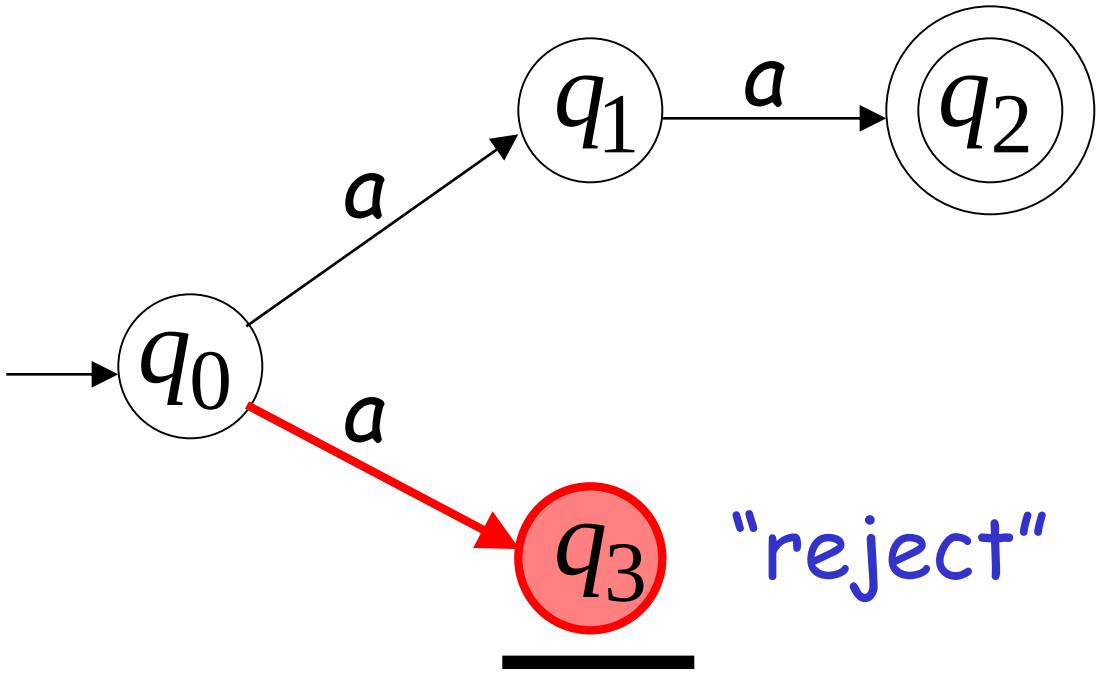
Second Choice





Second Choice





An NFA rejects a string:

when there is no computation of the NFA that accepts the string.

For each computation:

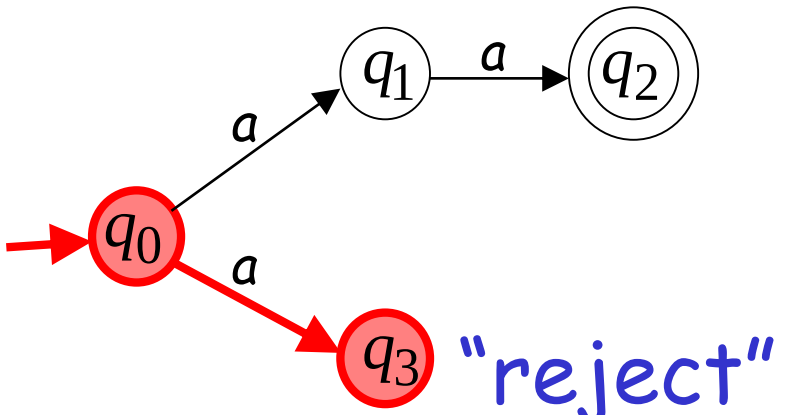
- All the input is consumed and the automaton is in a non final state

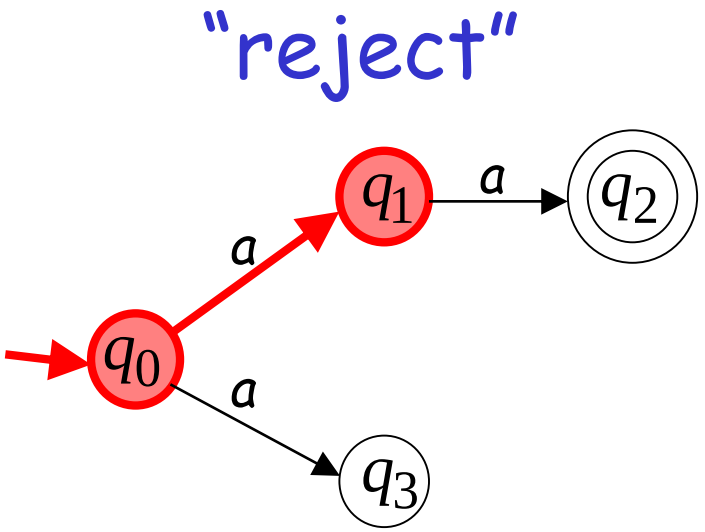
OR

- The input cannot be consumed

Example

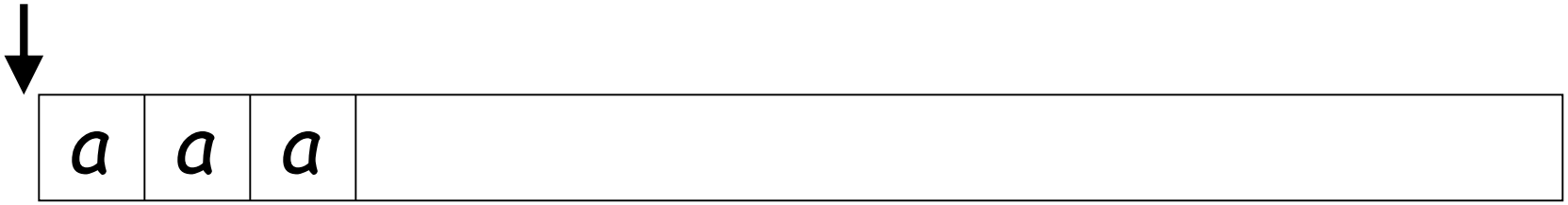
a is rejected by the NFA:

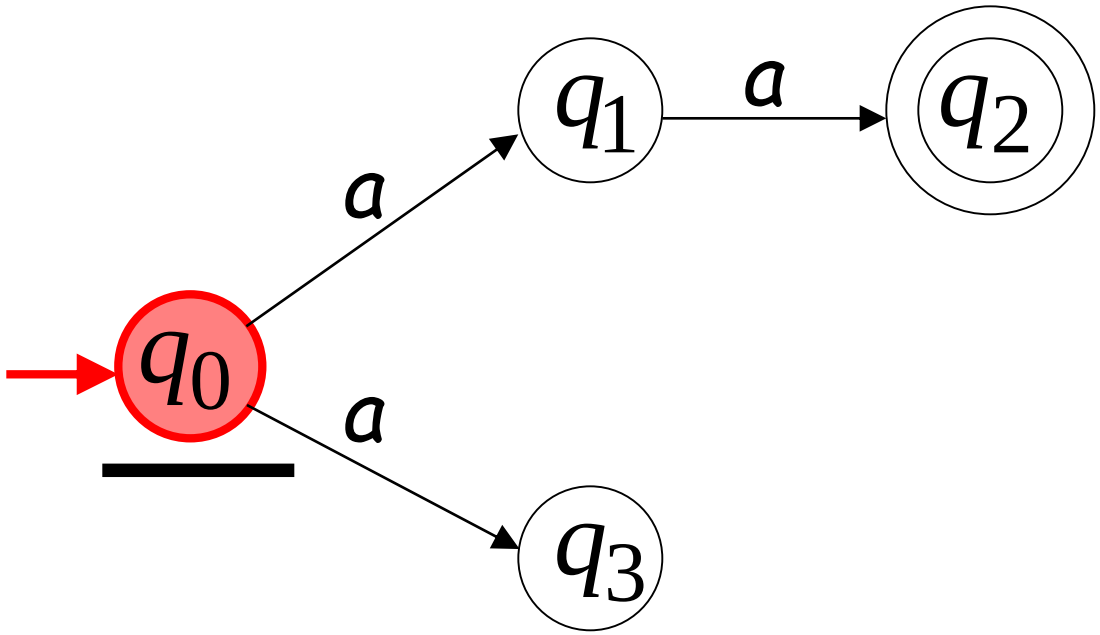




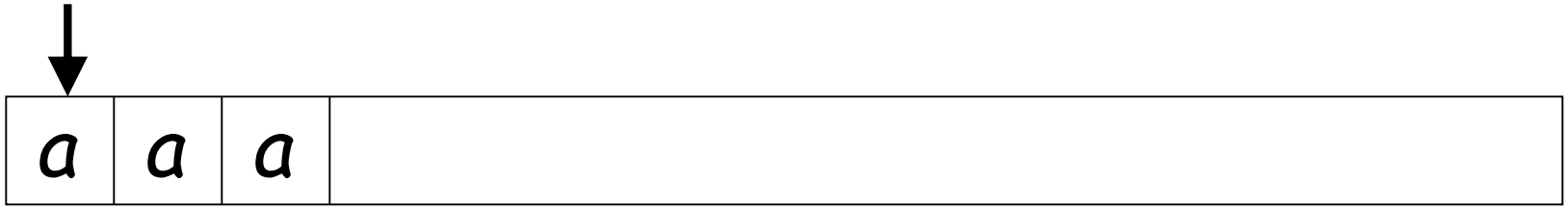
All possible computations lead to rejection

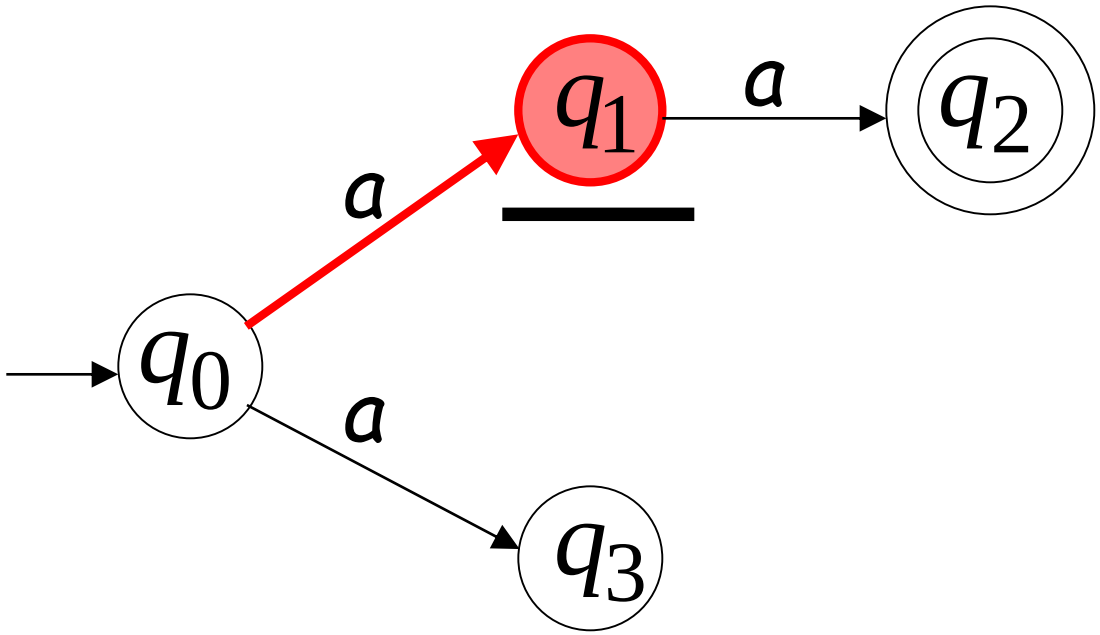
Rejection example



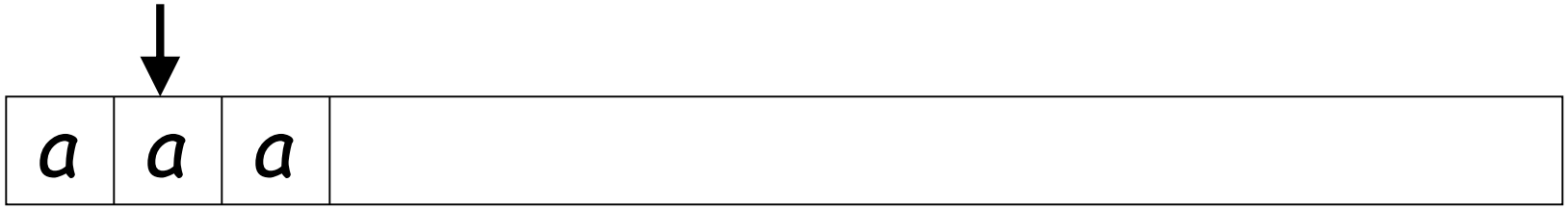


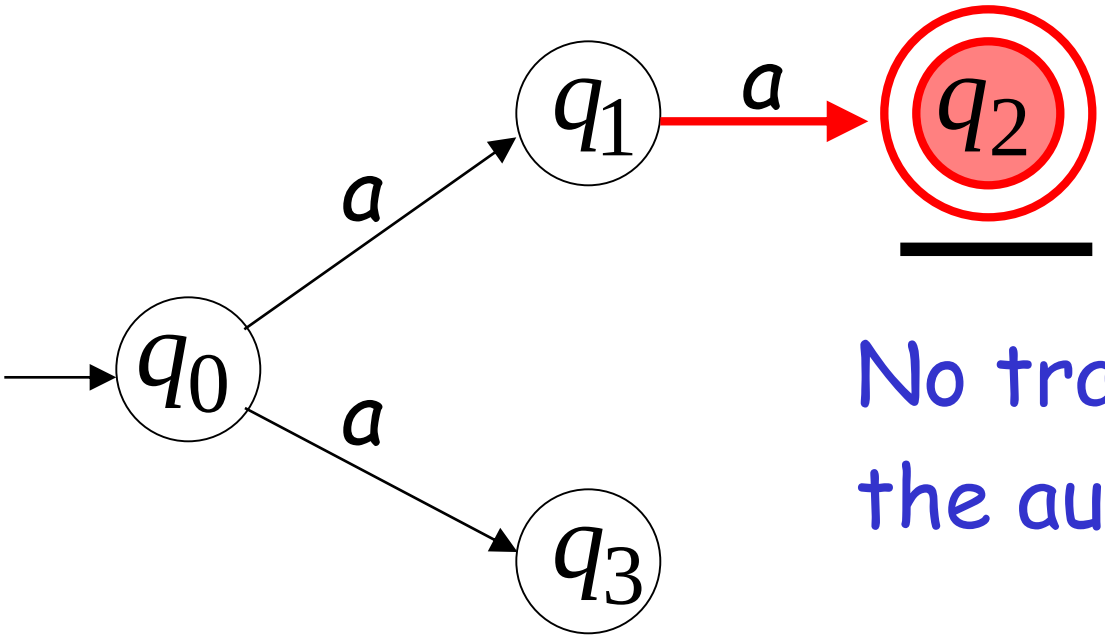
First Choice





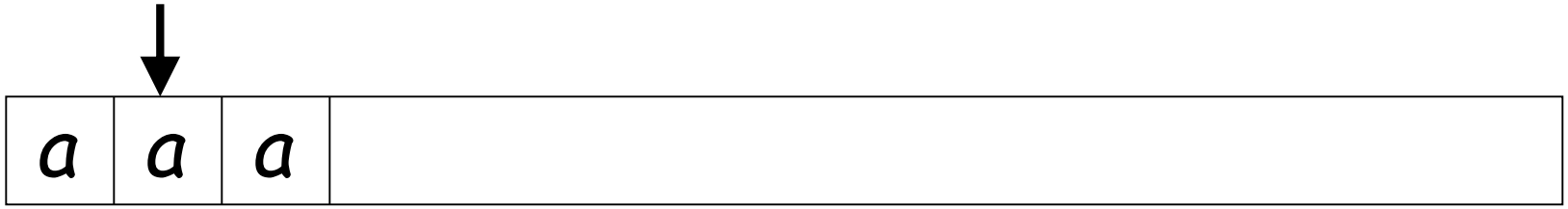
First Choice



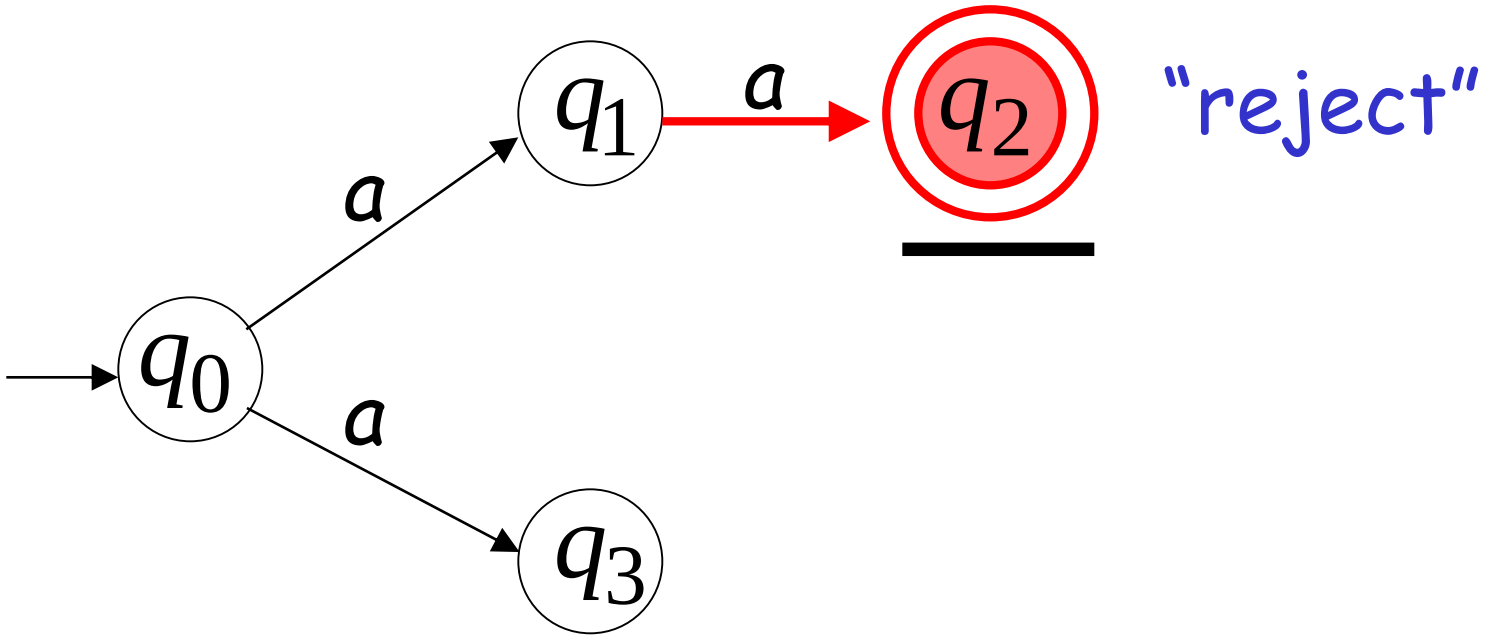


No transition:
the automaton hangs

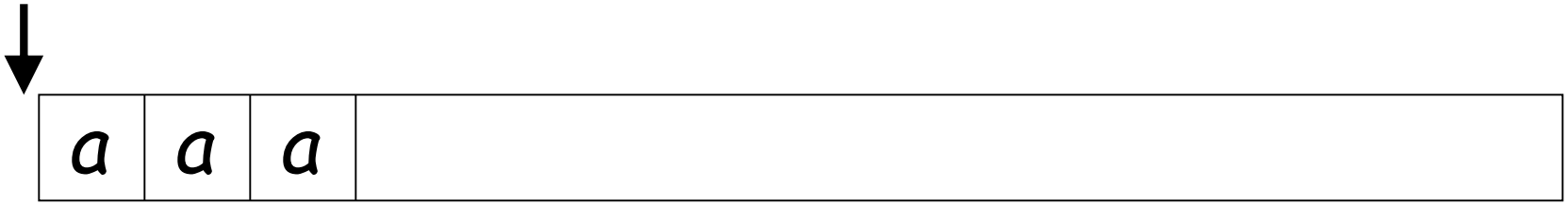
First Choice

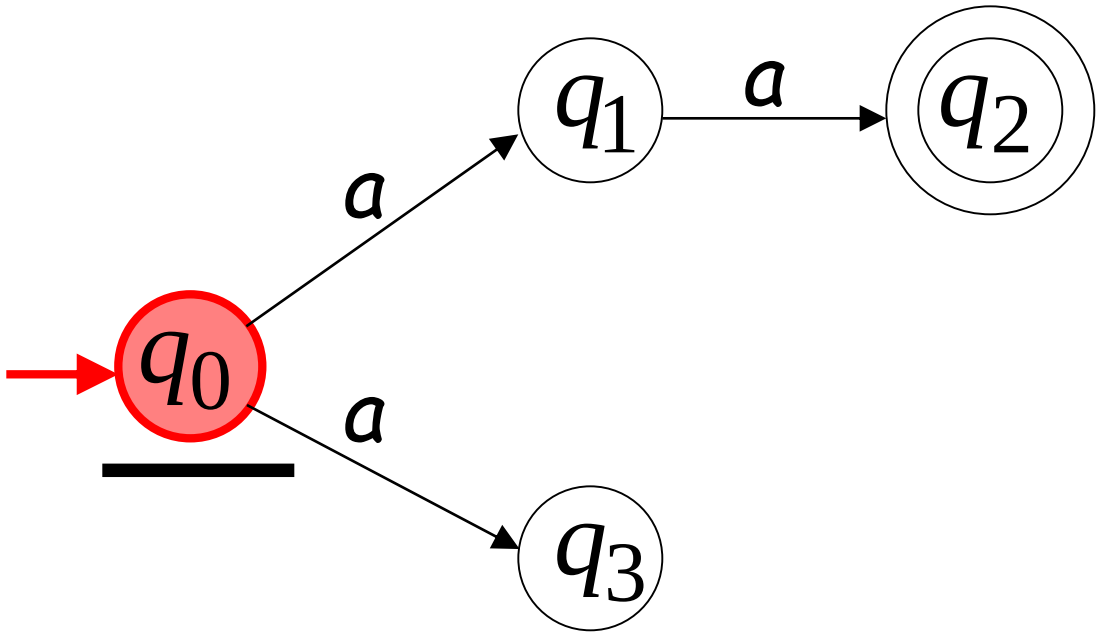


Input cannot be consumed

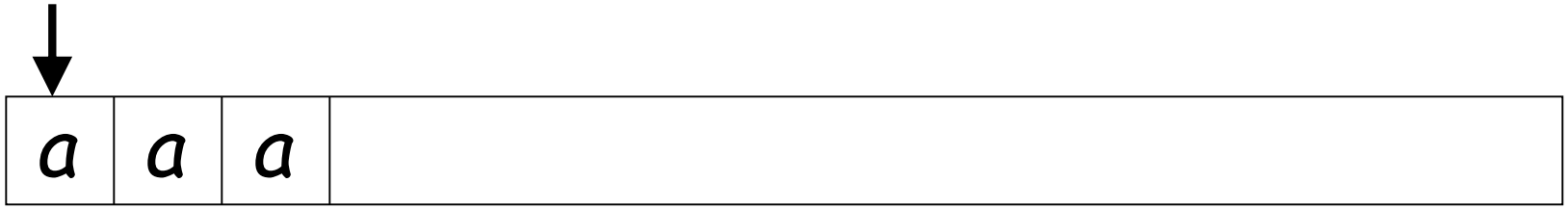


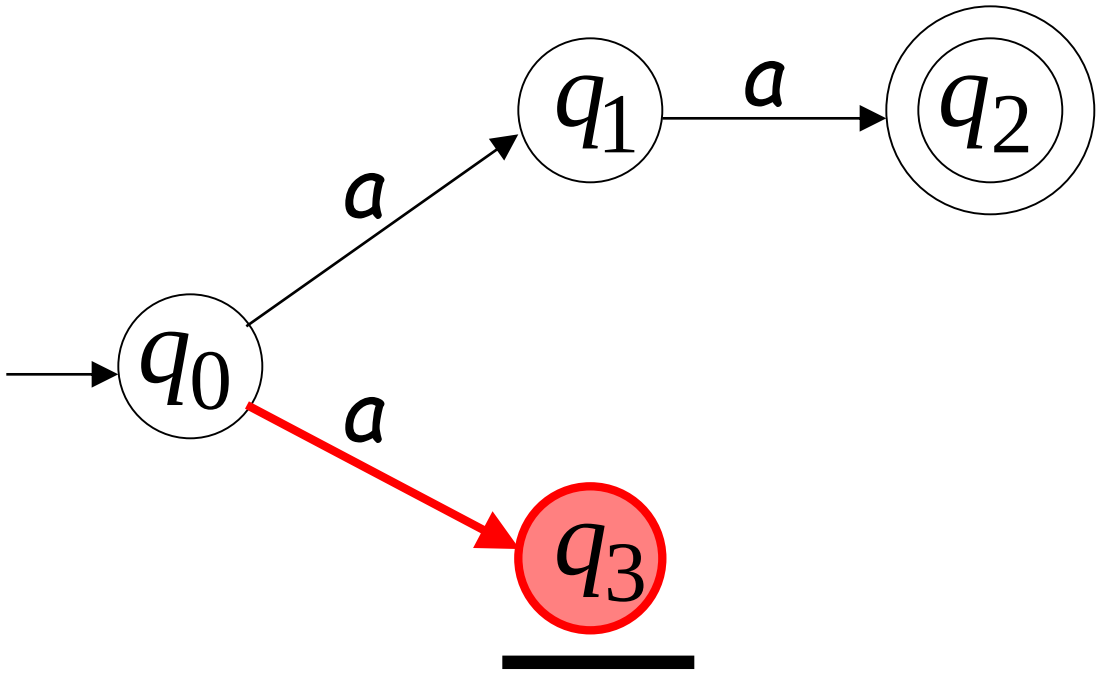
Second Choice



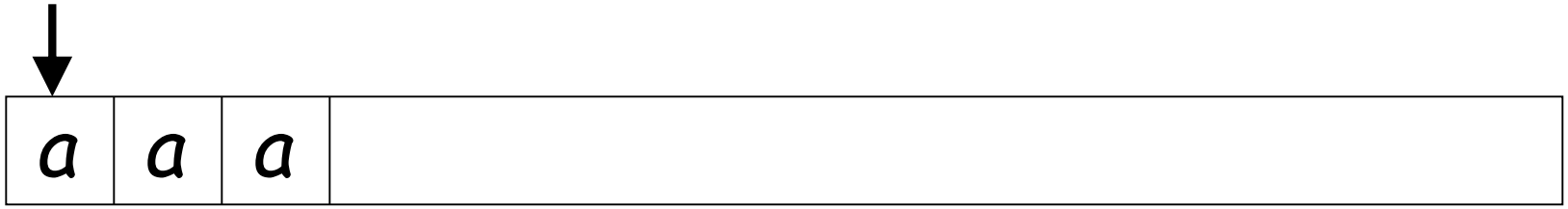


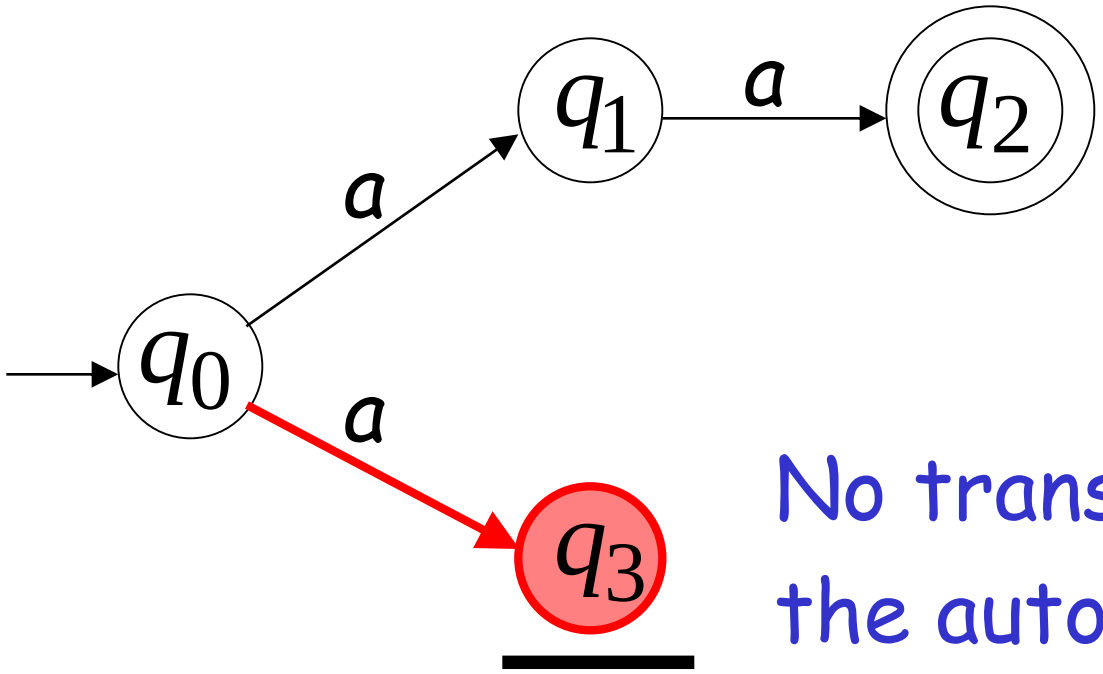
Second Choice





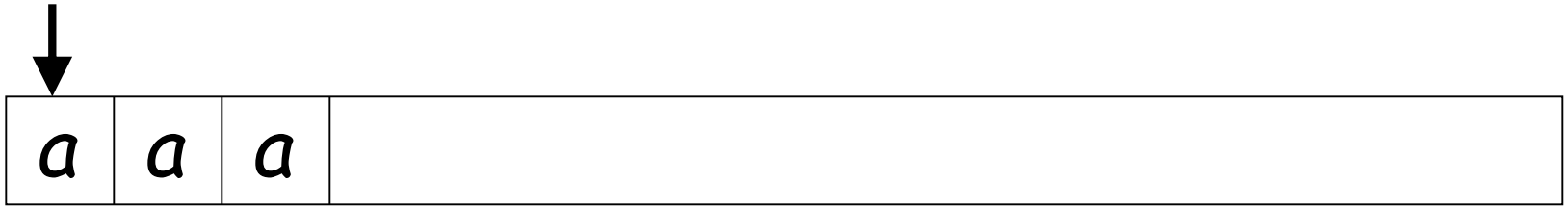
Second Choice



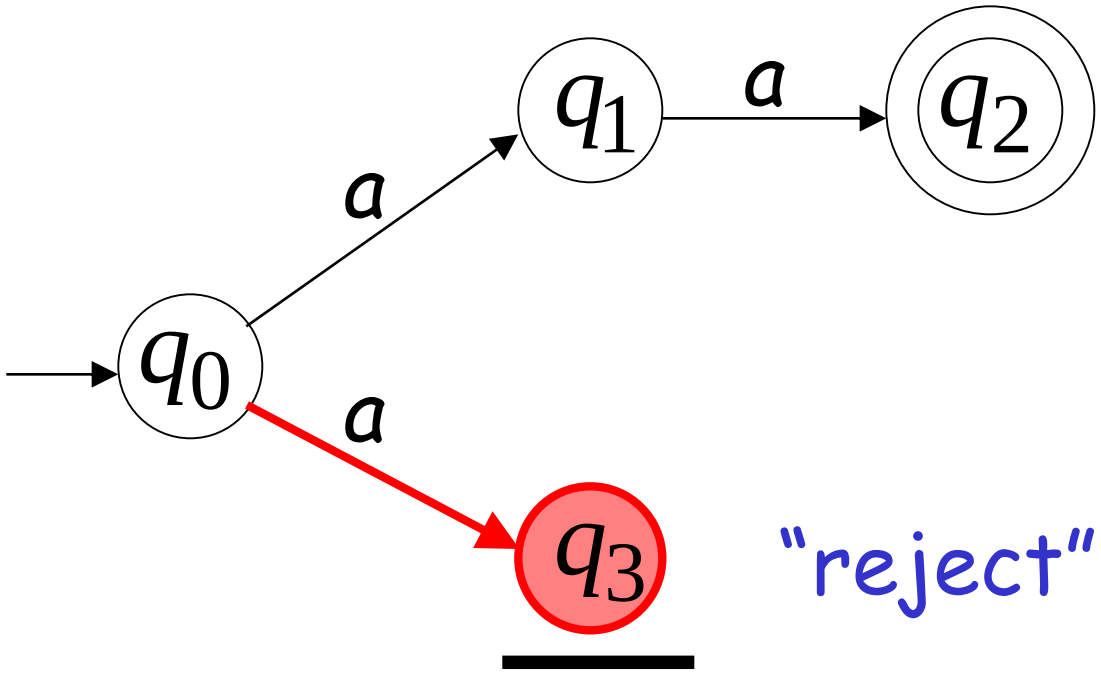


No transition:
the automaton hangs

Second Choice

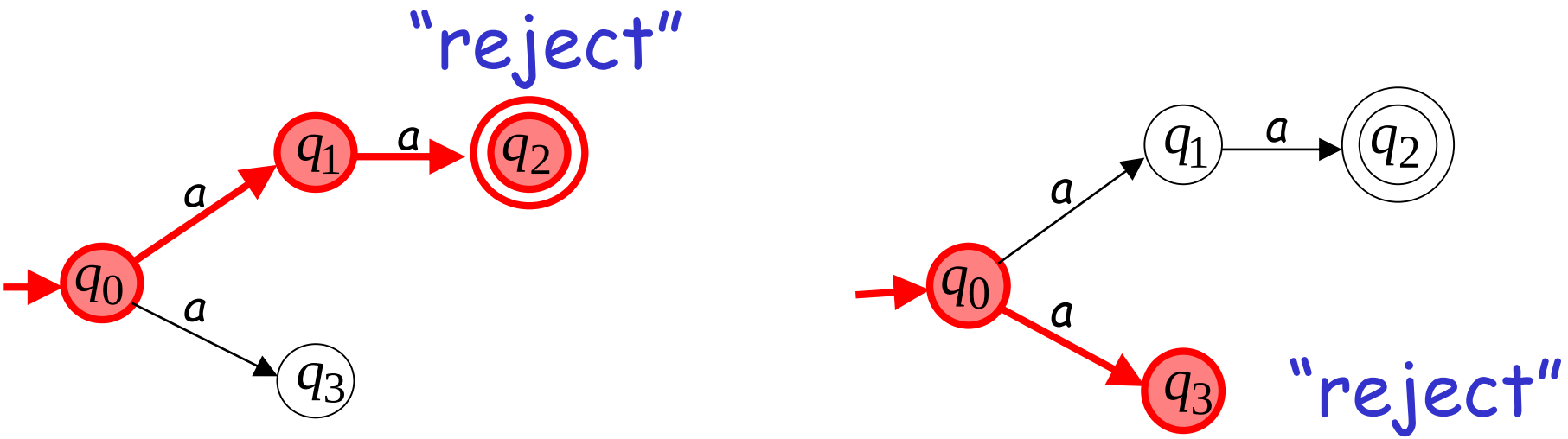


Input cannot be consumed



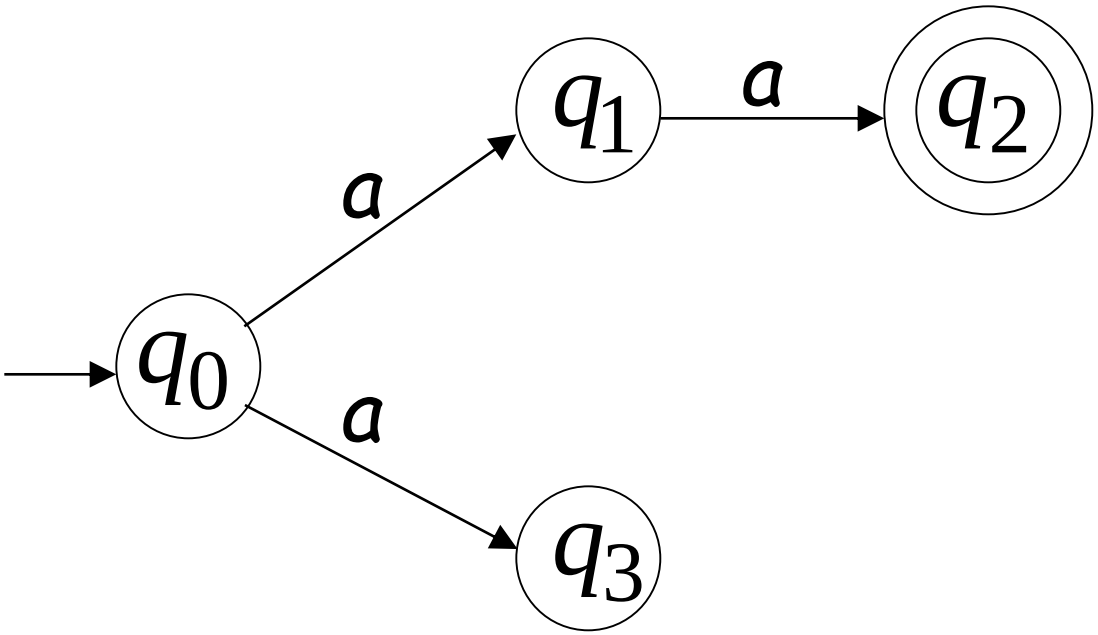
"reject"

aaa is rejected by the NFA:

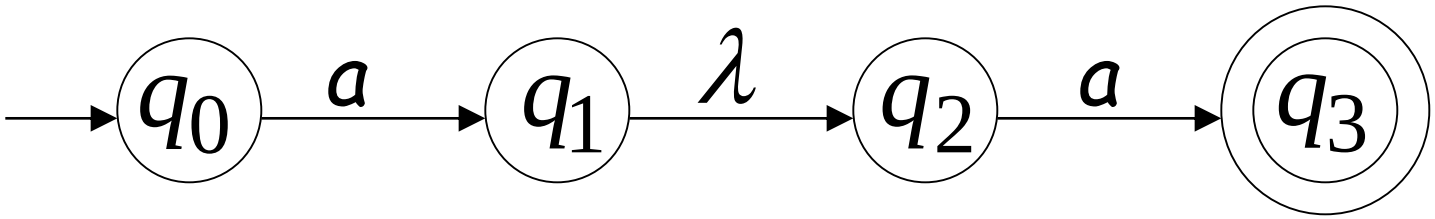


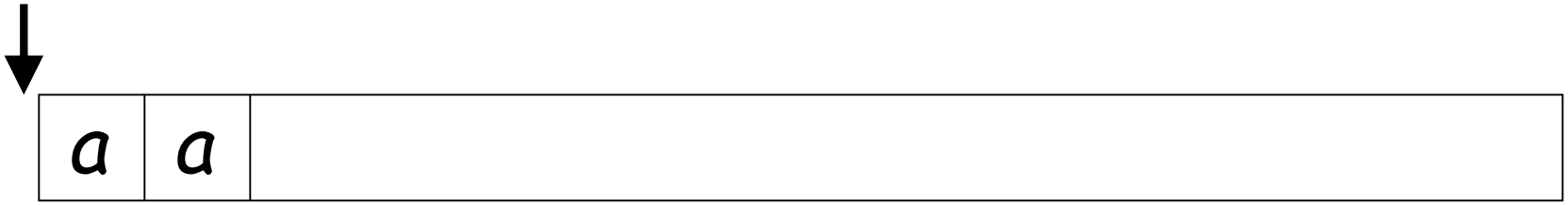
All possible computations lead to rejection

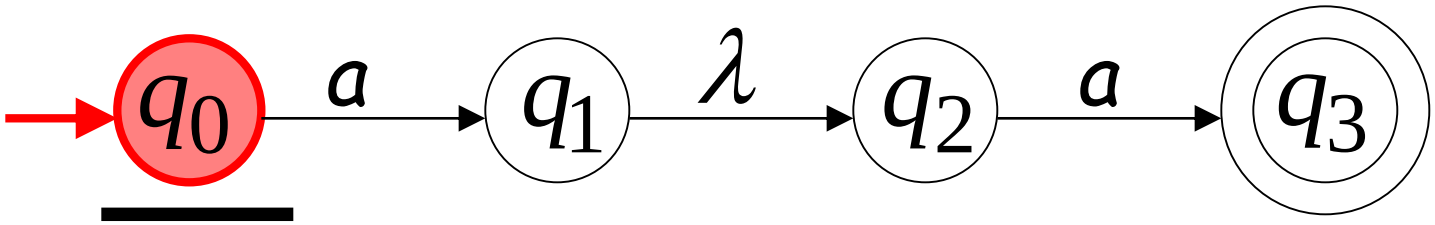
Language accepted: $L = \{aa\}$

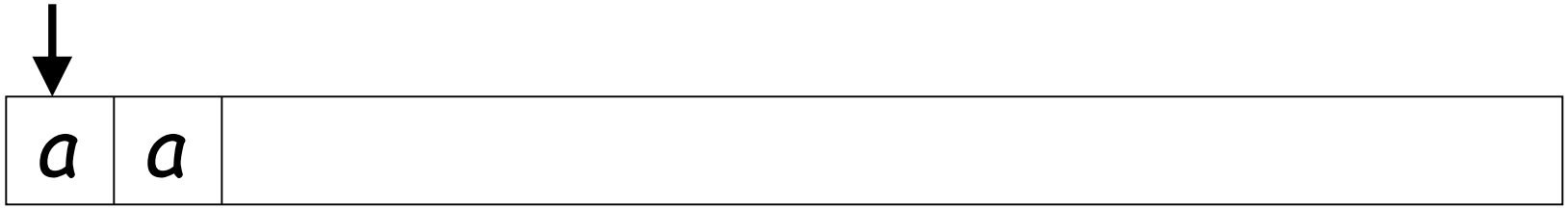


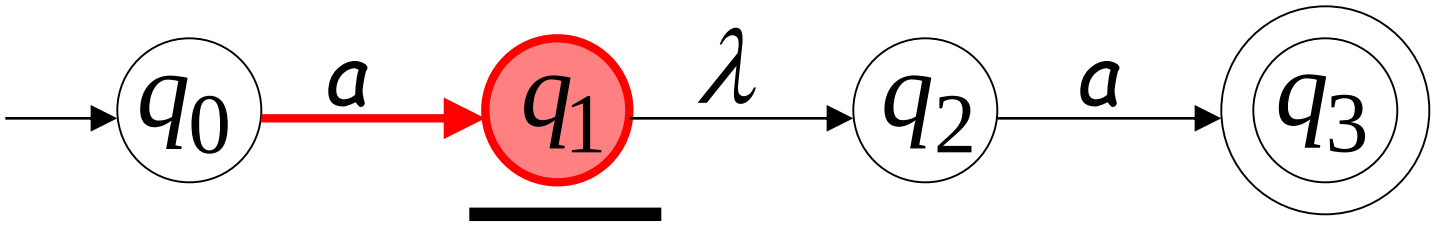
Lambda Transitions



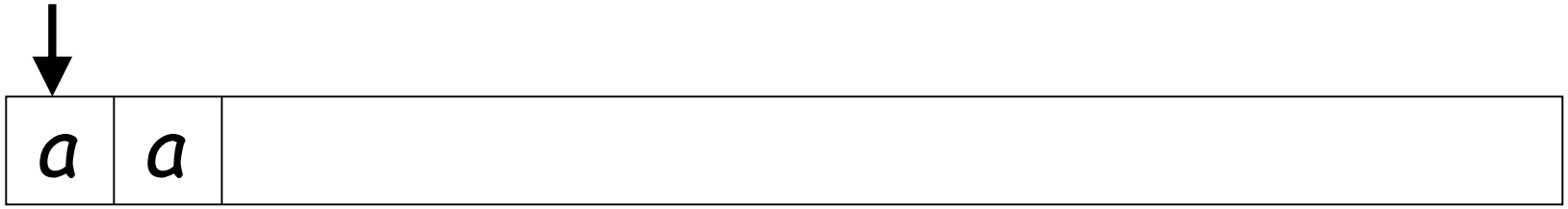

a
a

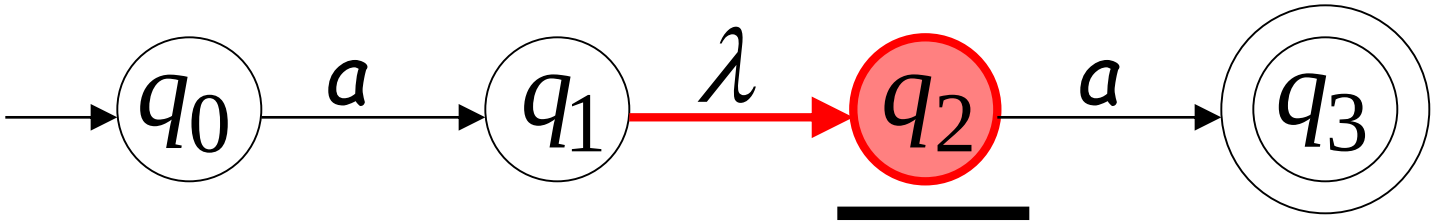

q_0
a
q_1
λ
q_2
a
q_3

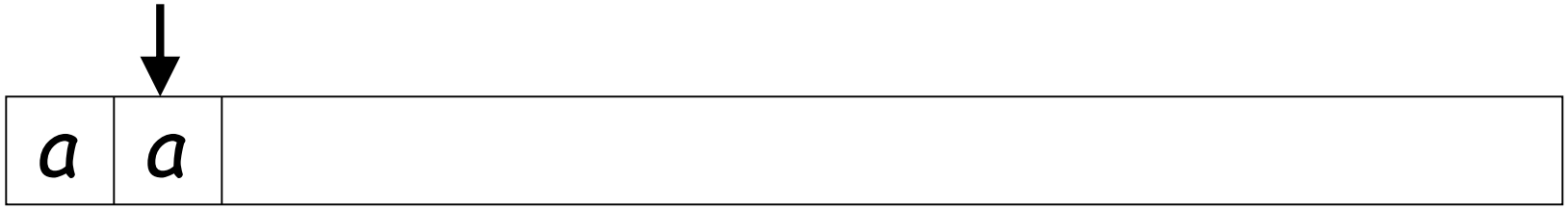

a
a

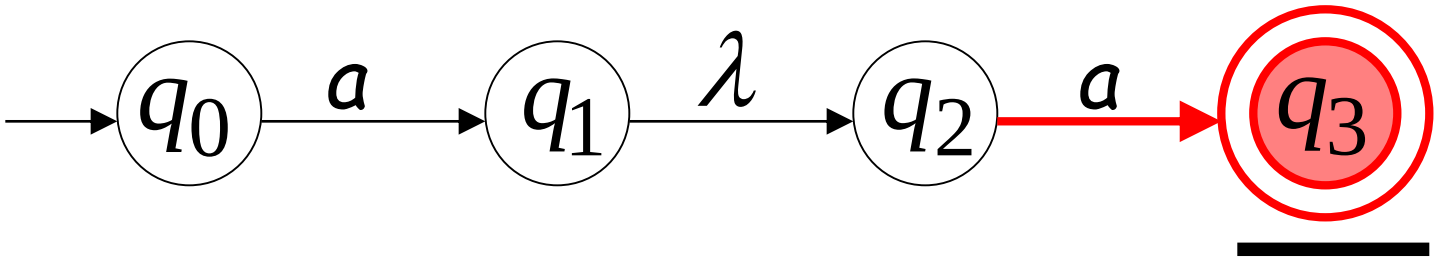

q_0
a
q_1
λ
q_2
a
q_3

(read head does not move)

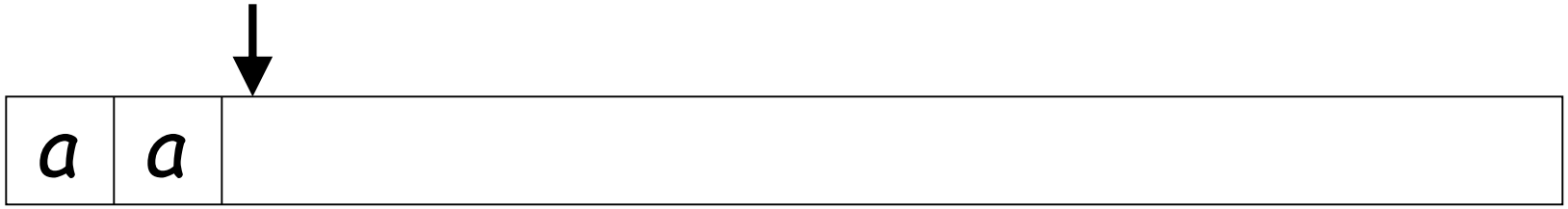




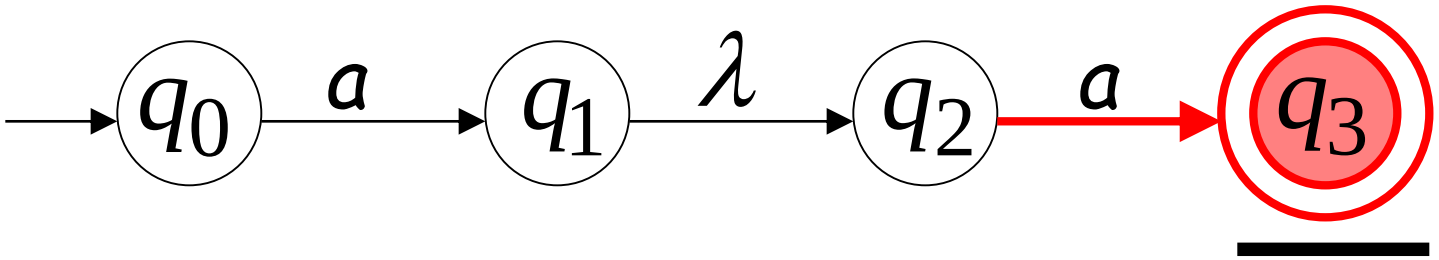

a
a


q_0
a
q_1
λ
q_2
a
q_3

all input is consumed

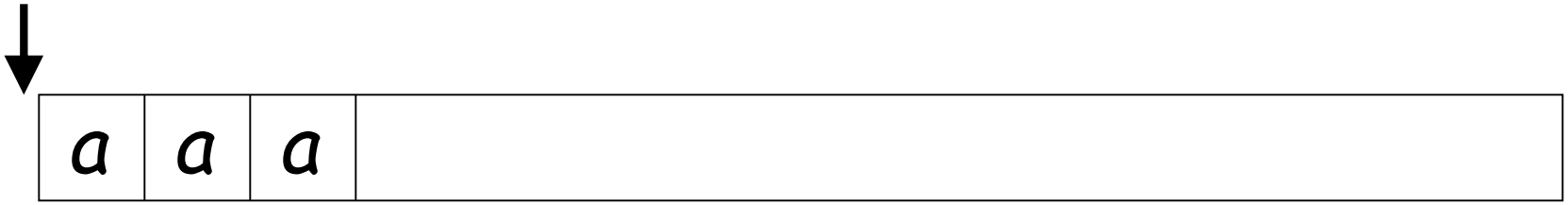


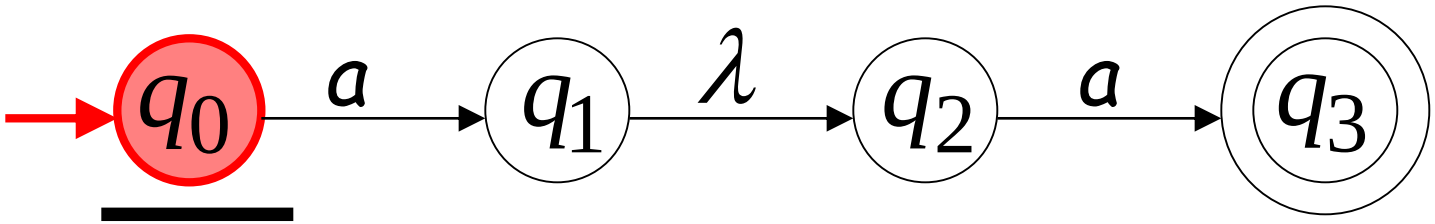
"accept"

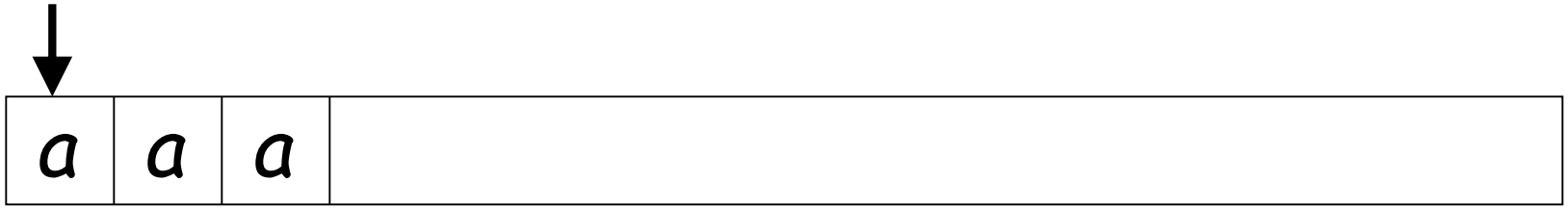


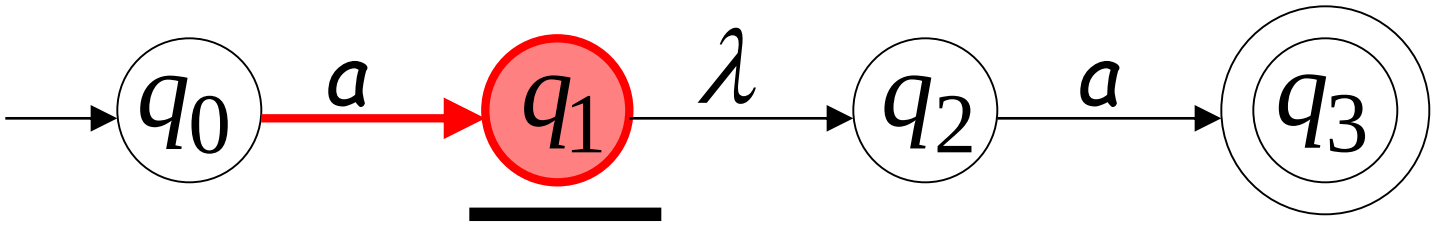
String aa is accepted

Rejection Example

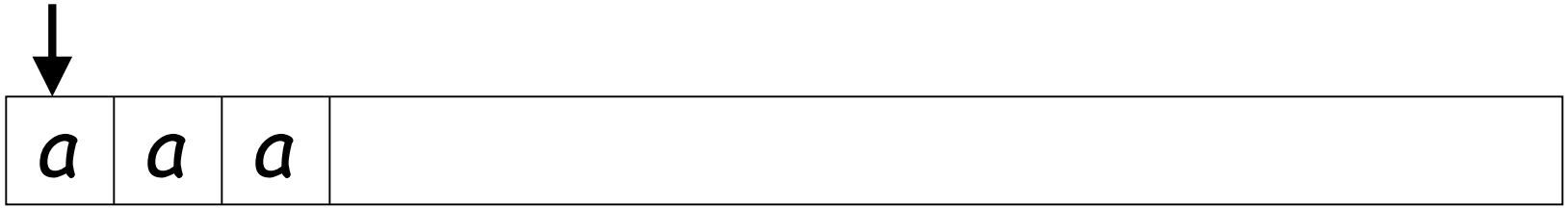


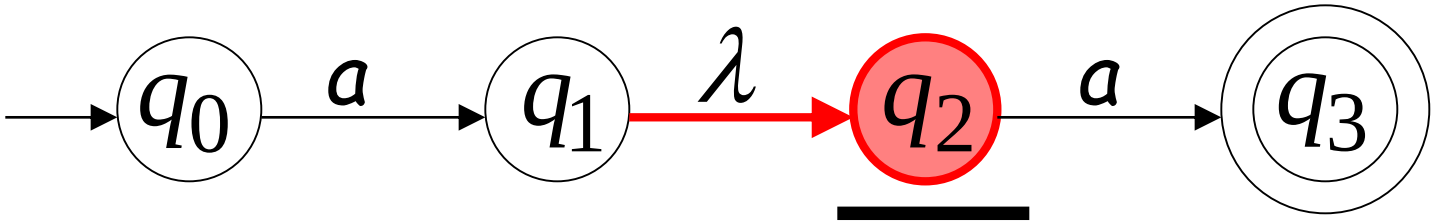


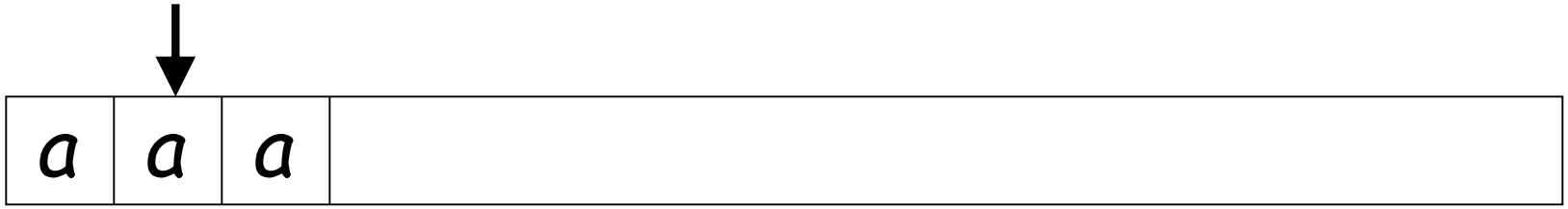

a
a
a

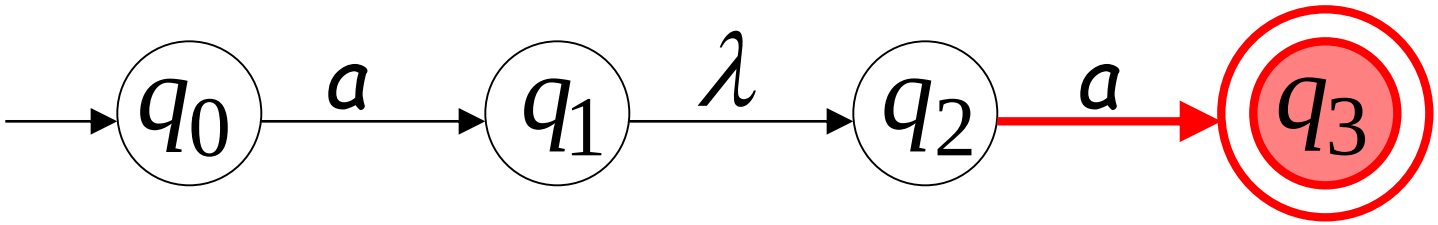

q_0
a
q_1
λ
q_2
a
q_3

(read head doesn't move)





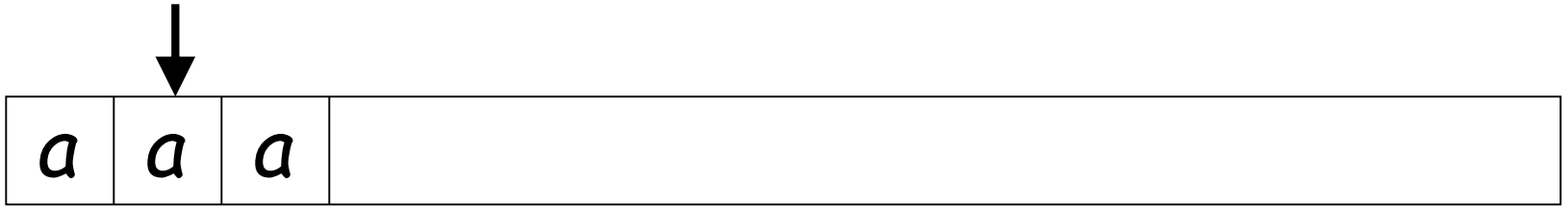




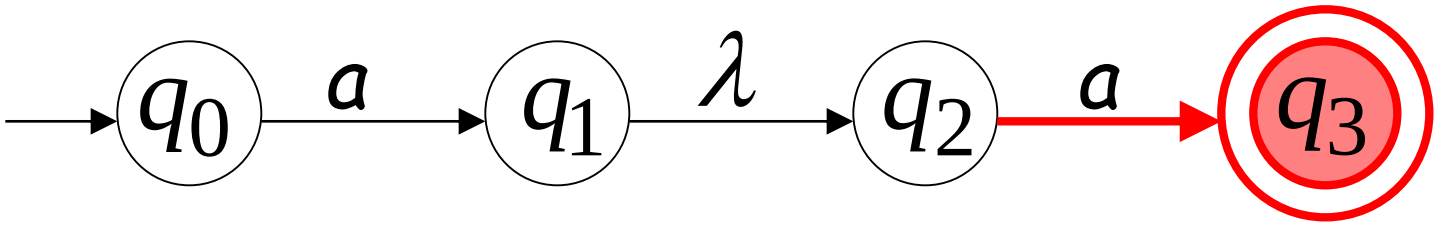
No transition:

the automaton hangs

Input cannot be consumed

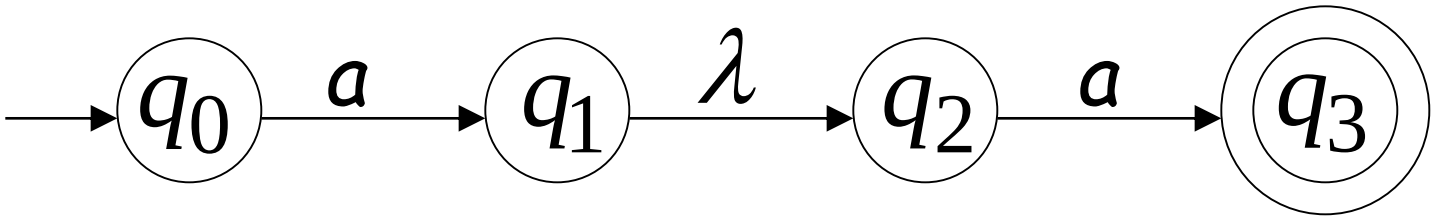


"reject"

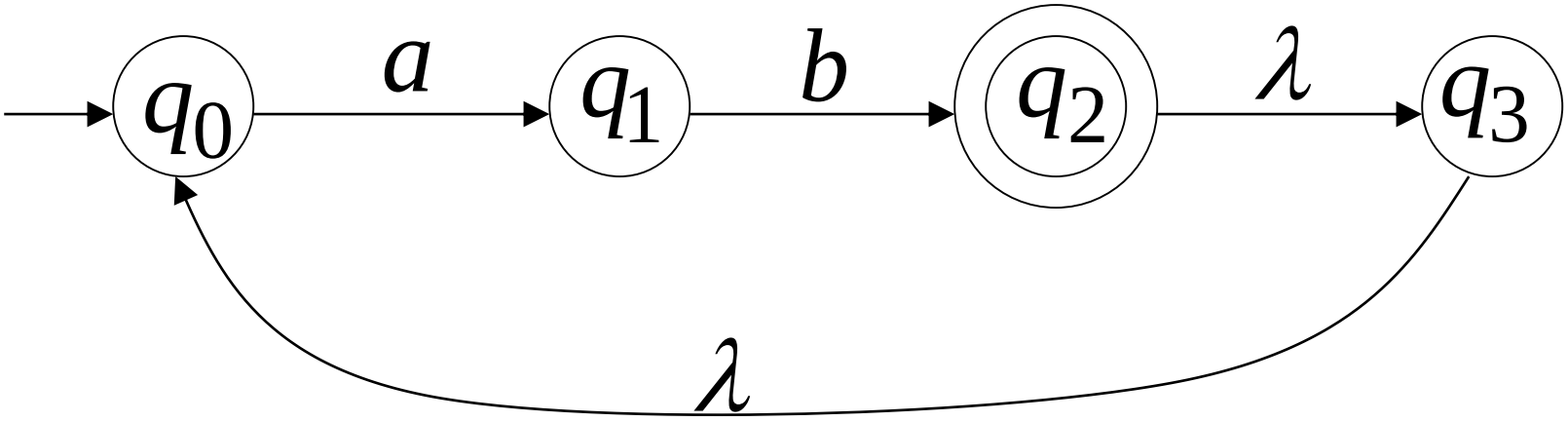


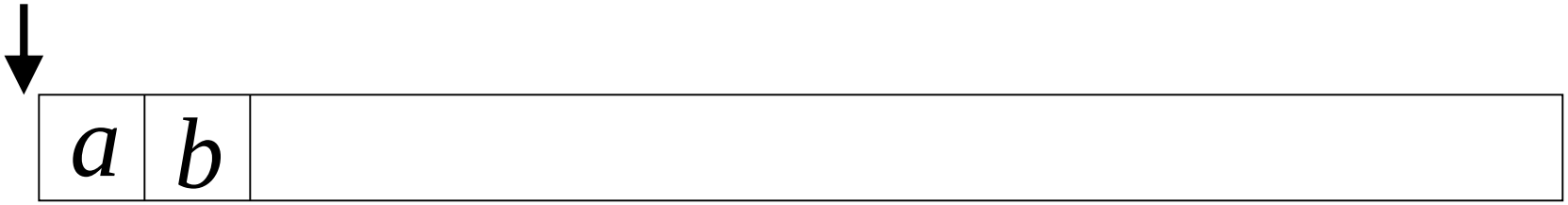
String aaa is rejected

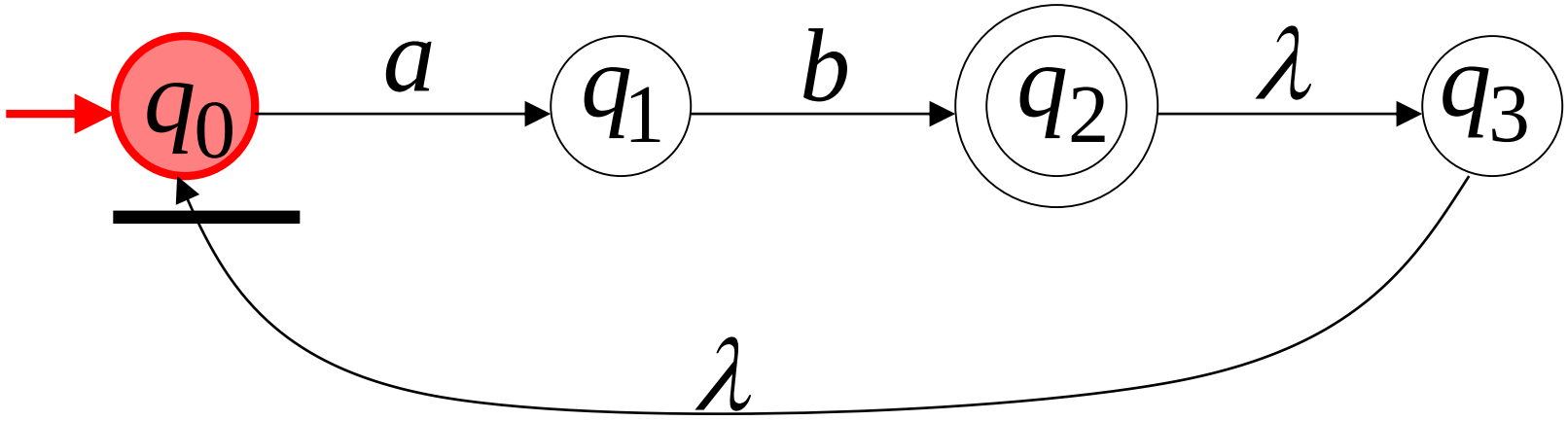
Language accepted: $L = \{aa\}$

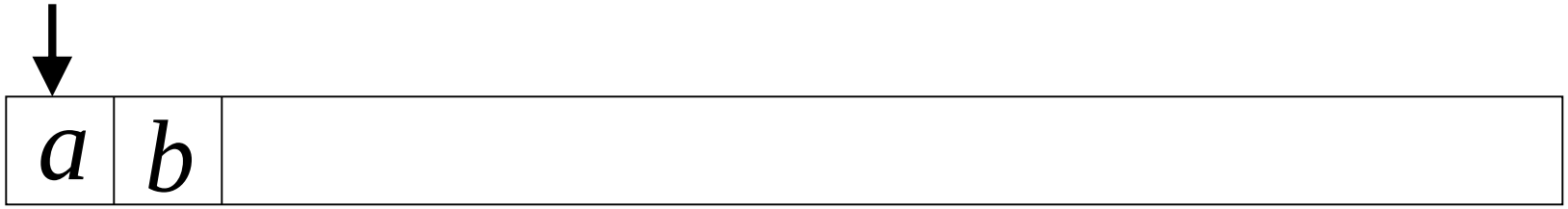


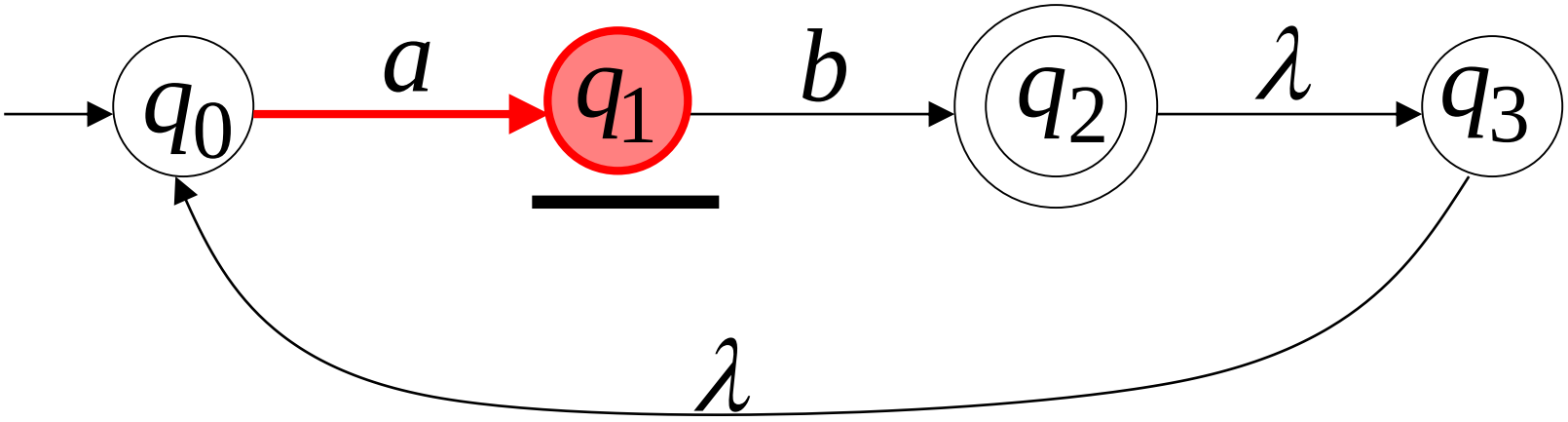
Another NFA Example

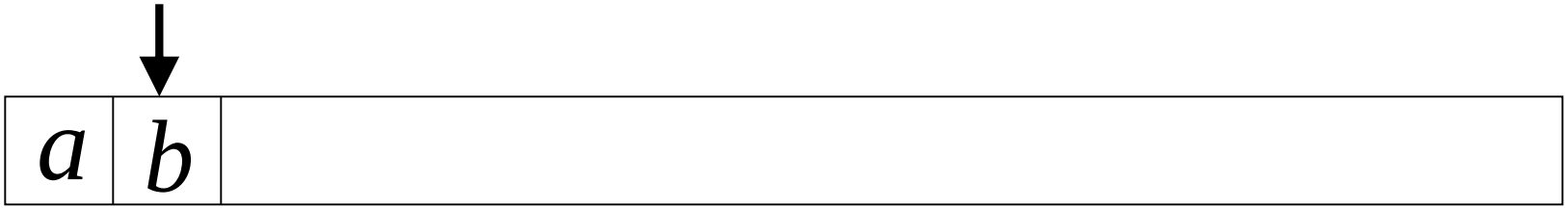


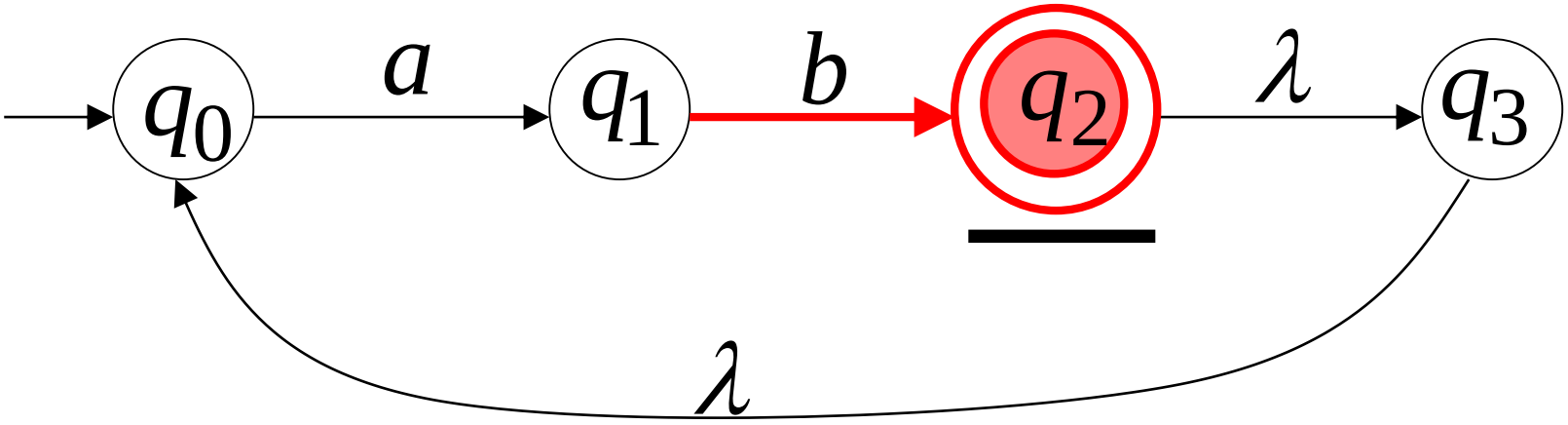

a
b

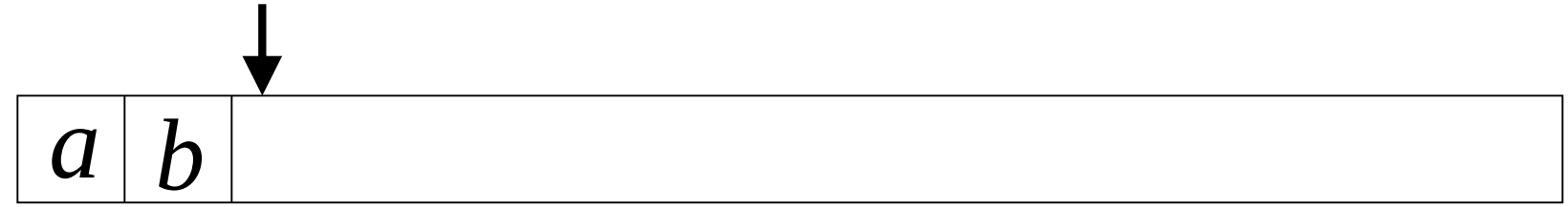

q_0
a
q_1
b
q_2
λ
q_3
λ

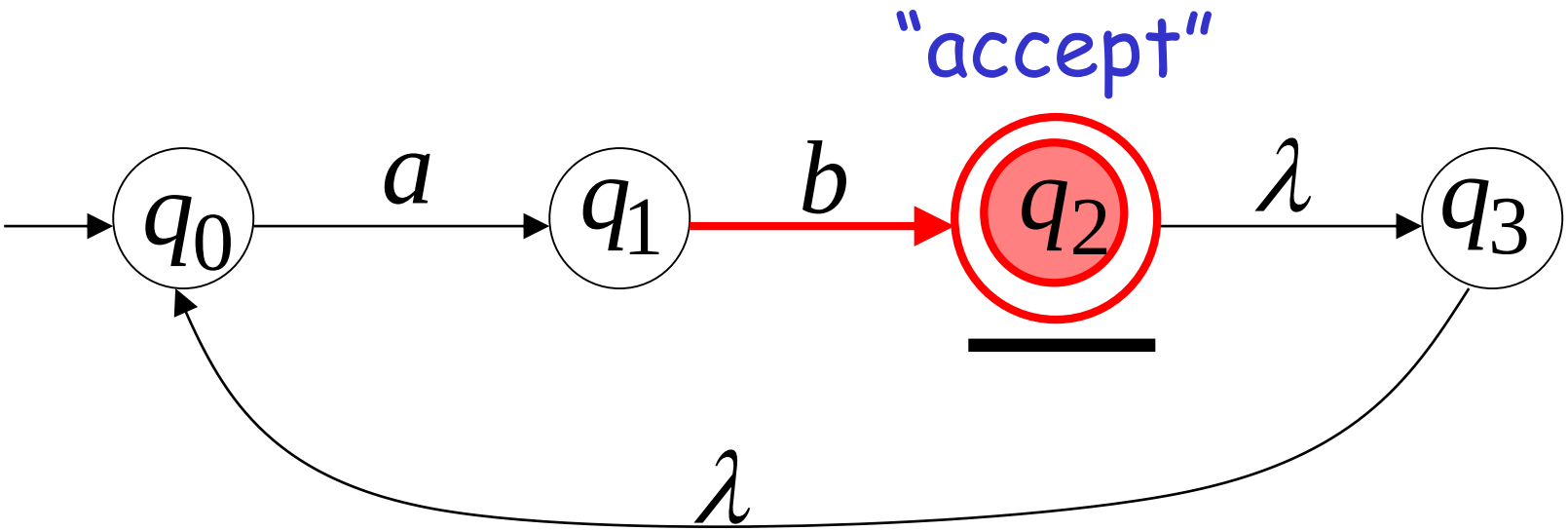

a
b


q_0
a
q_1
b
q_2
λ
q_3
λ

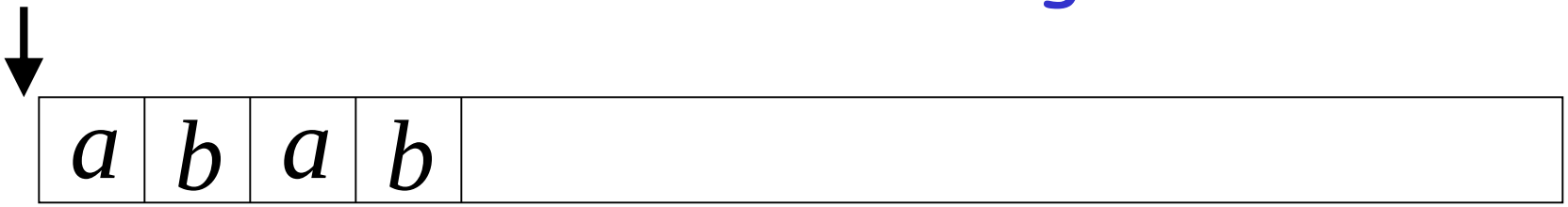

a
b

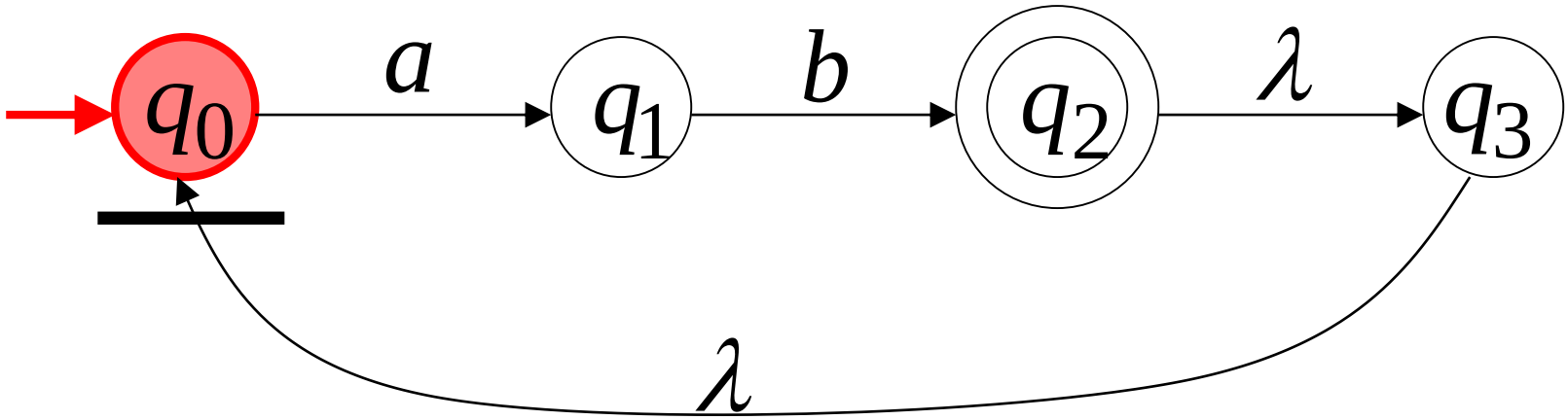

q_0
a
q_1
b
q_2
λ
q_3
λ

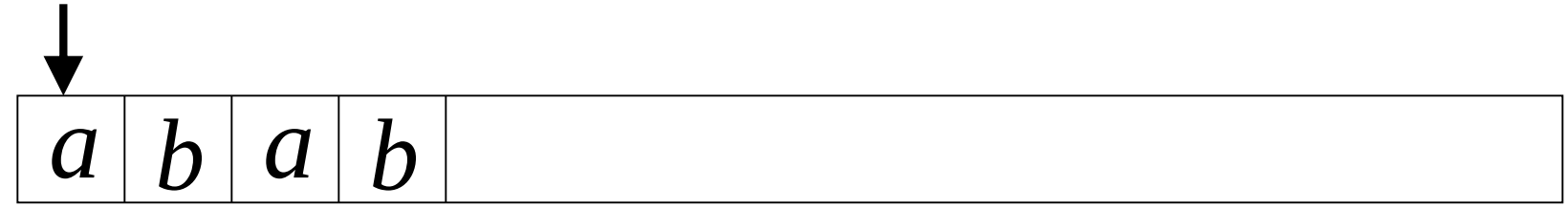

a
b

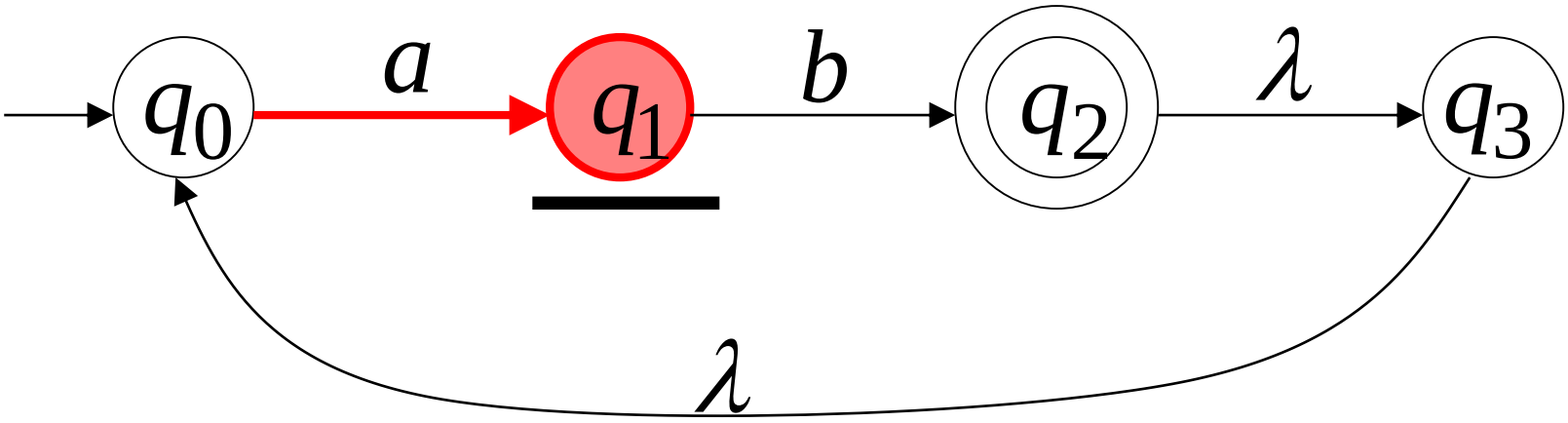

"accept"
q_0
q_1
q_2
q_3
a
b
λ
λ

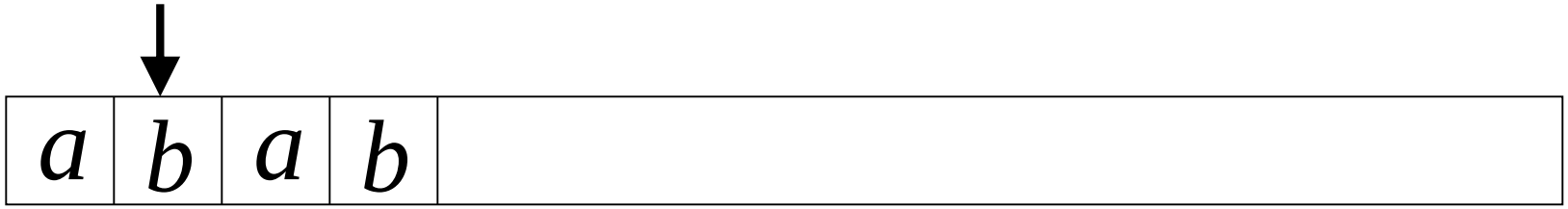
Another String

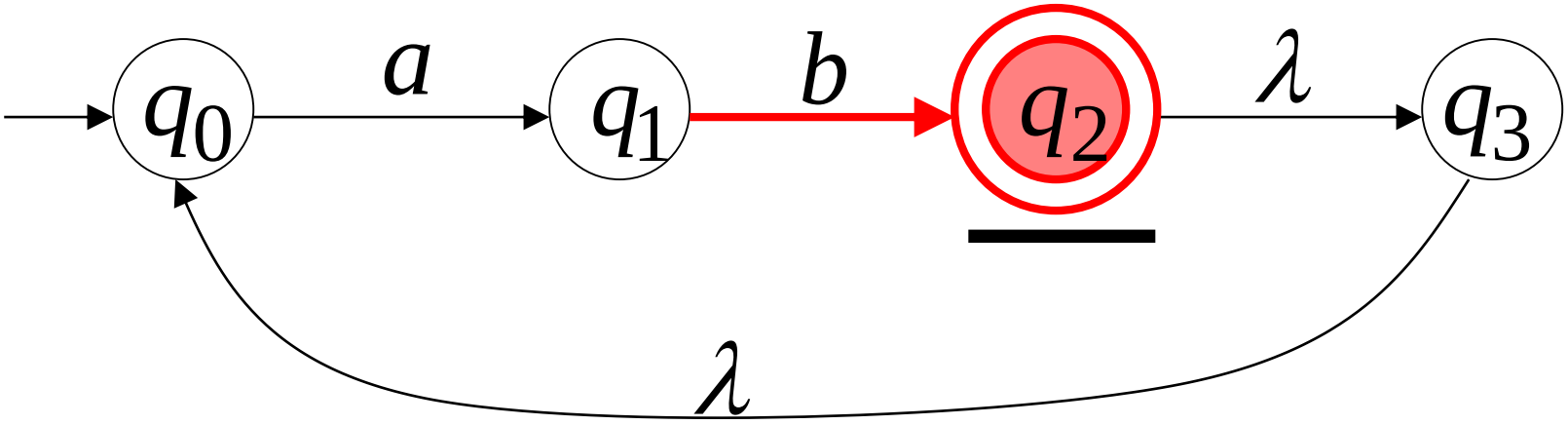


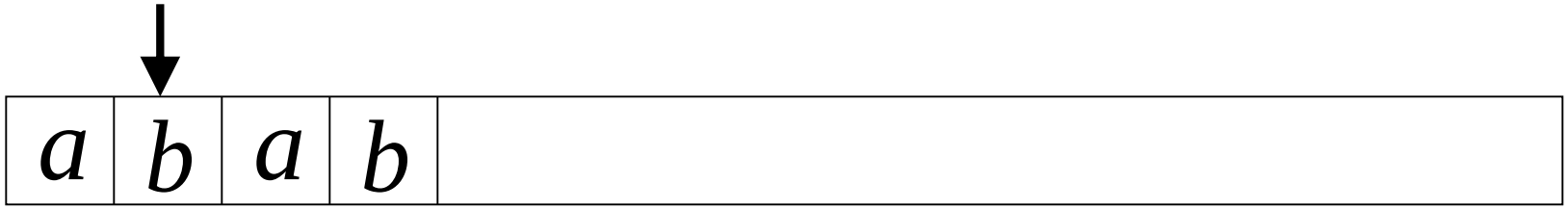


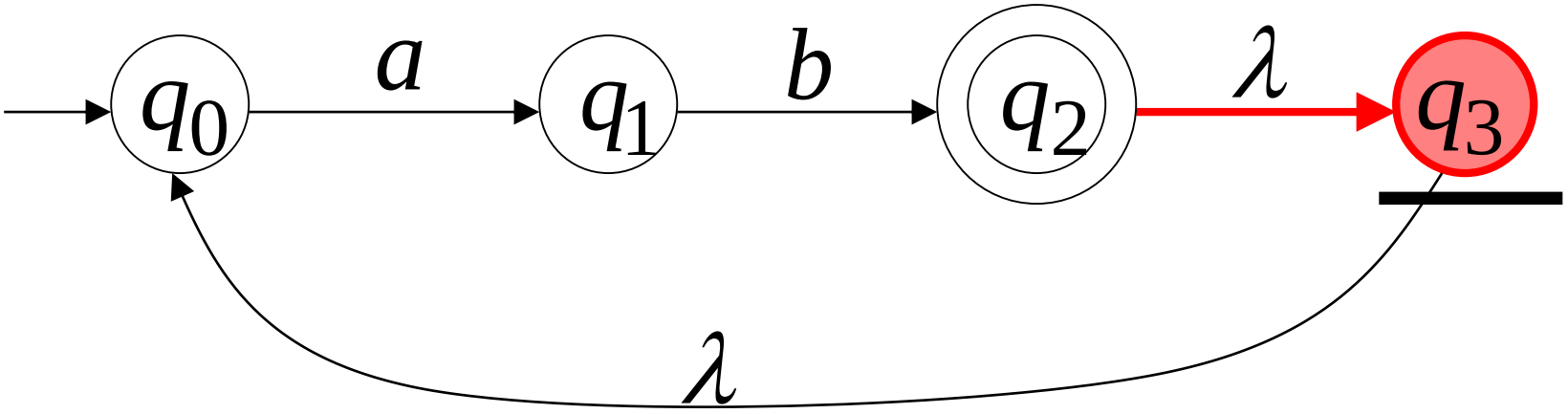

a
b
a
b

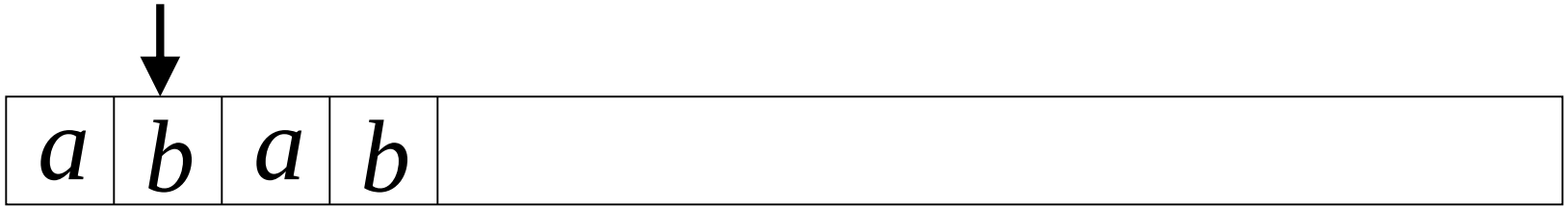

q_0
a
q_1
b
q_2
λ
q_3
λ

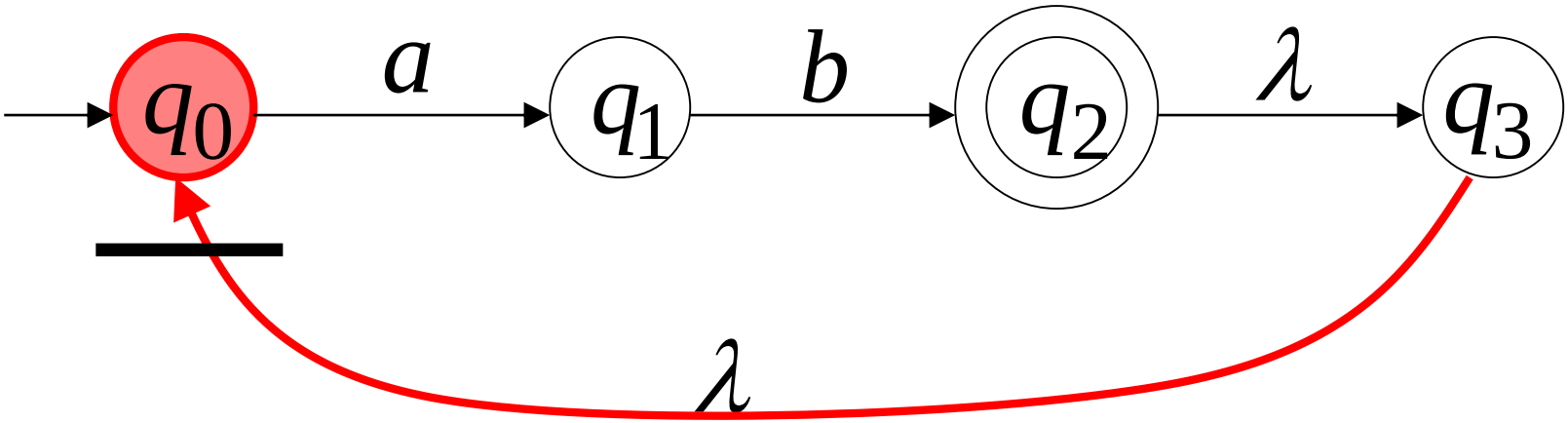

a
b
a
b

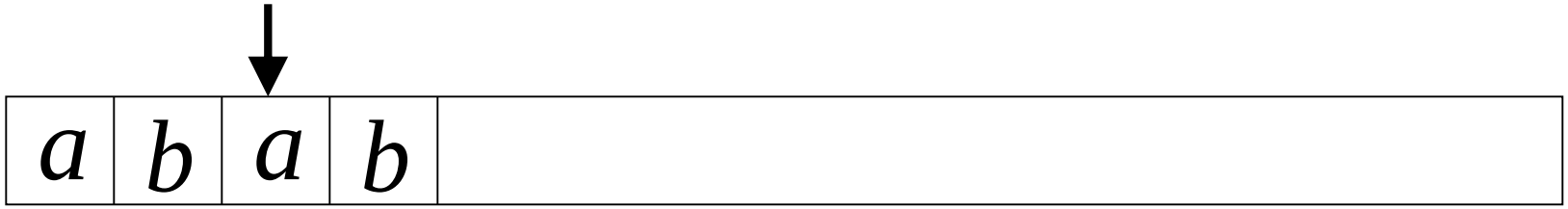

q_0
a
q_1
b
q_2
λ
q_3
λ

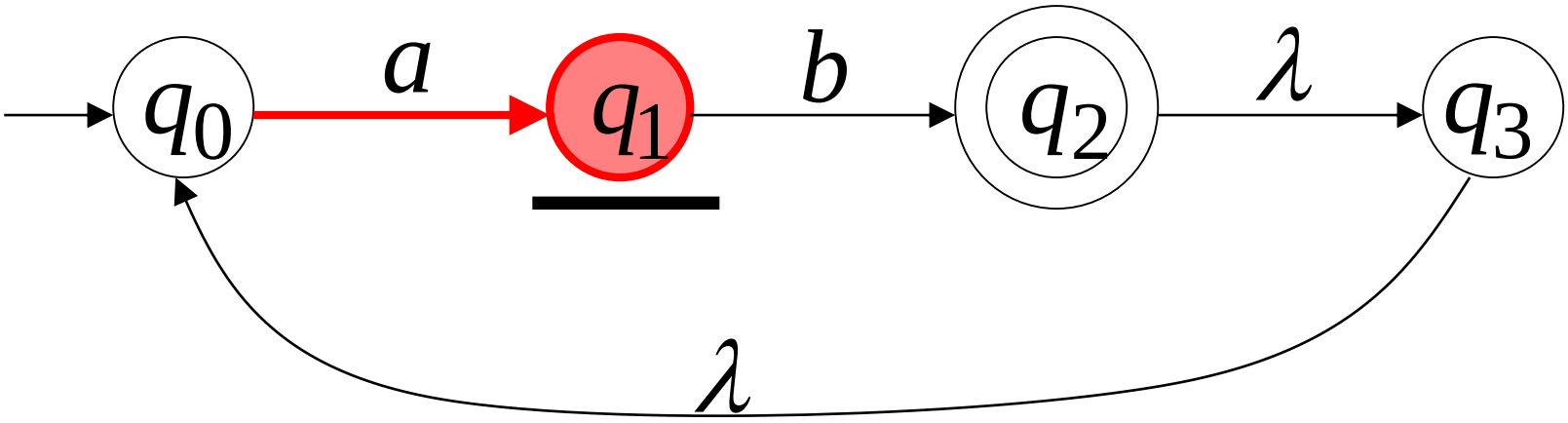

a
b
a
b

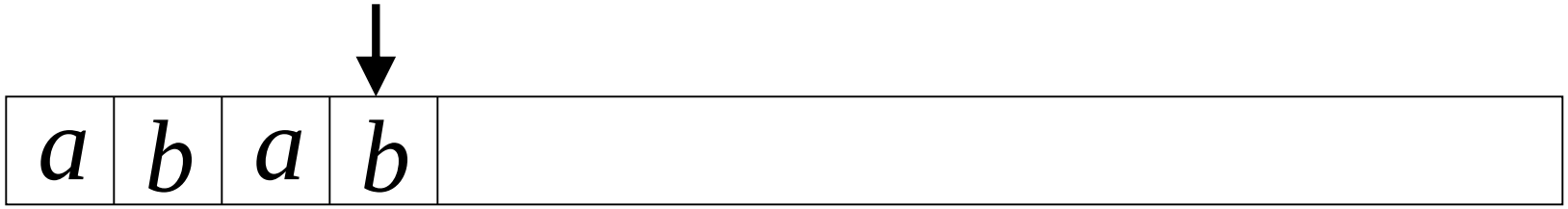

q_0
a
q_1
b
q_2
λ
q_3
λ

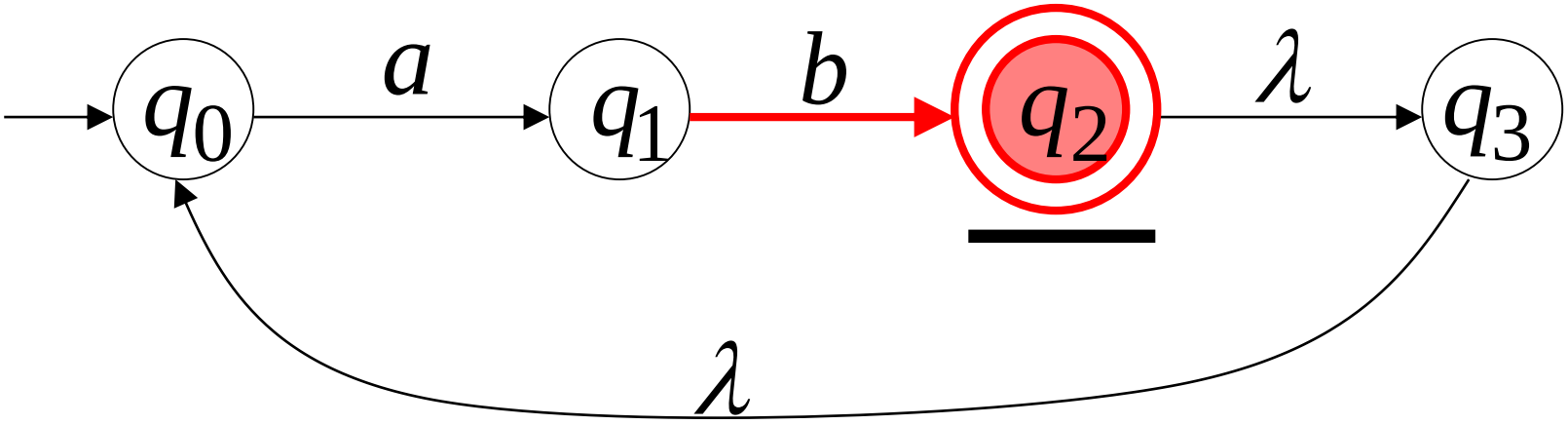

a
b
a
b

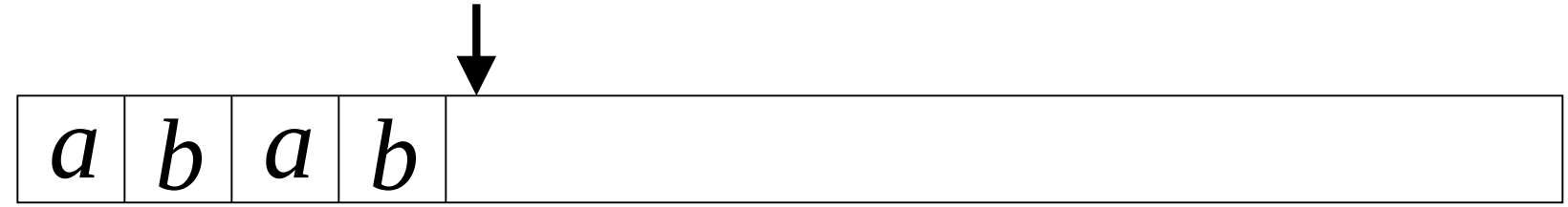

q_0
a
q_1
b
q_2
λ
q_3
λ

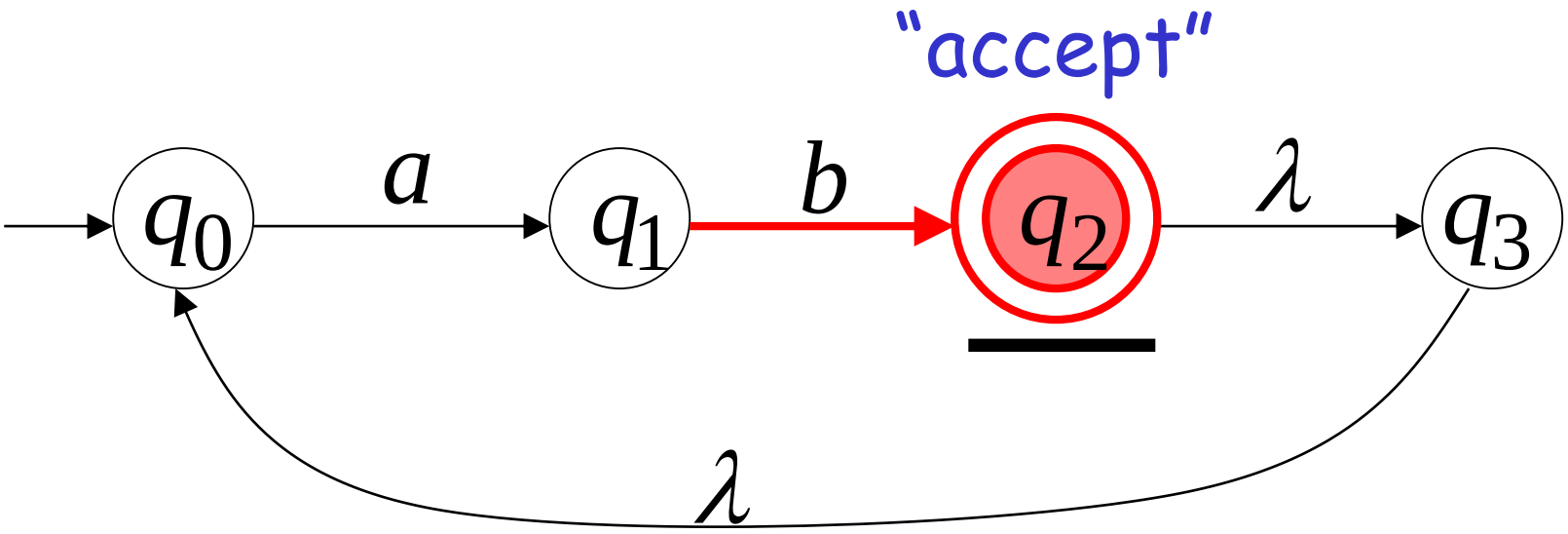

a
b
a
b


q_0
a
q_1
b
q_2
λ
q_3
λ


a
b
a
b


q_0
a
q_1
b
q_2
λ
q_3
λ

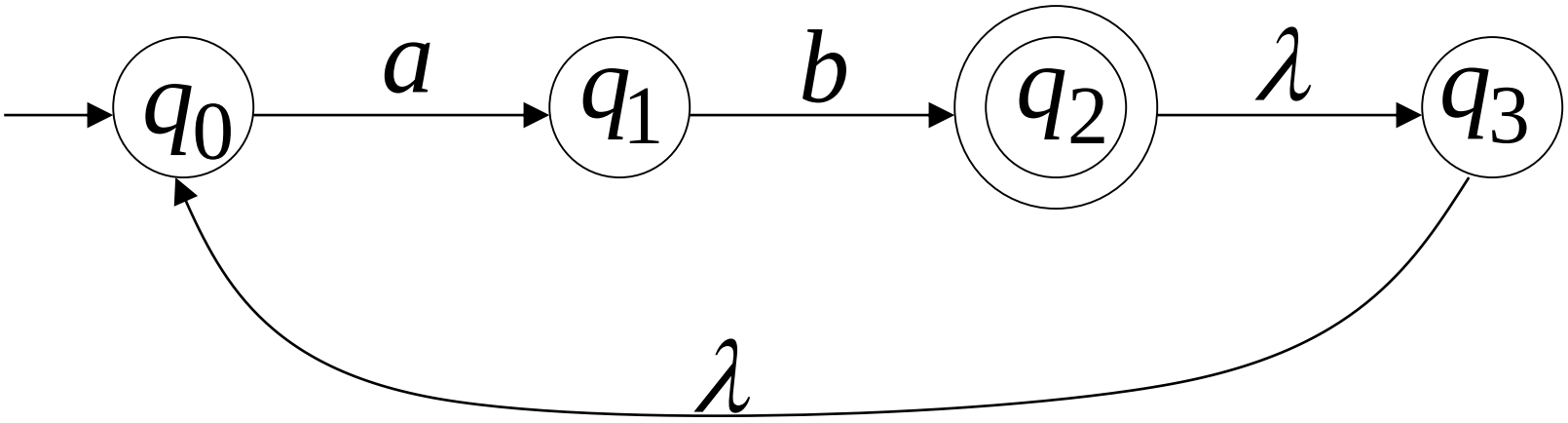

a
b
a
b


"accept"
q_0
q_1
q_2
q_3
a
b
λ
λ

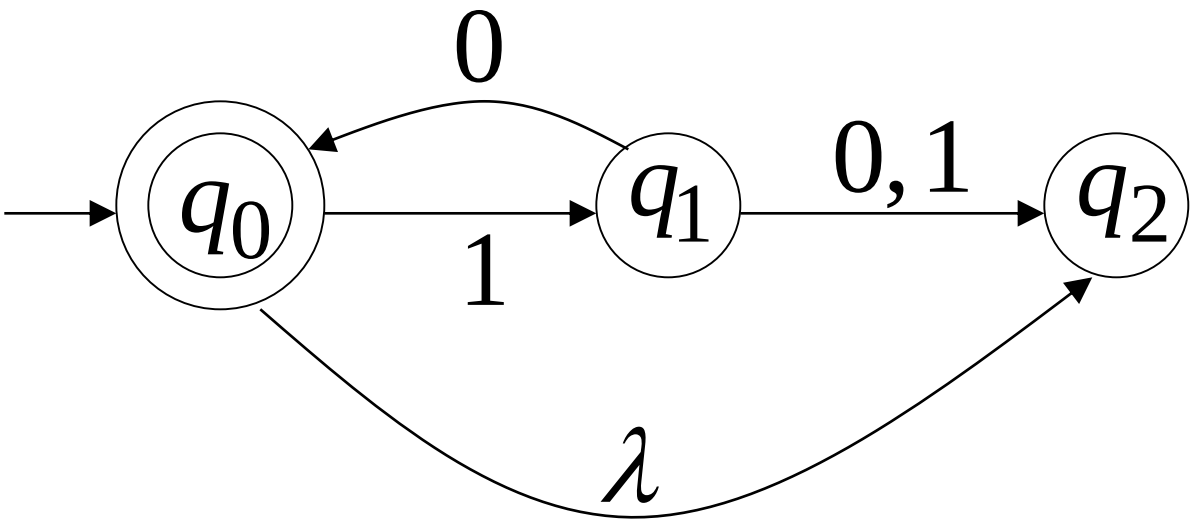
Language accepted

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

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



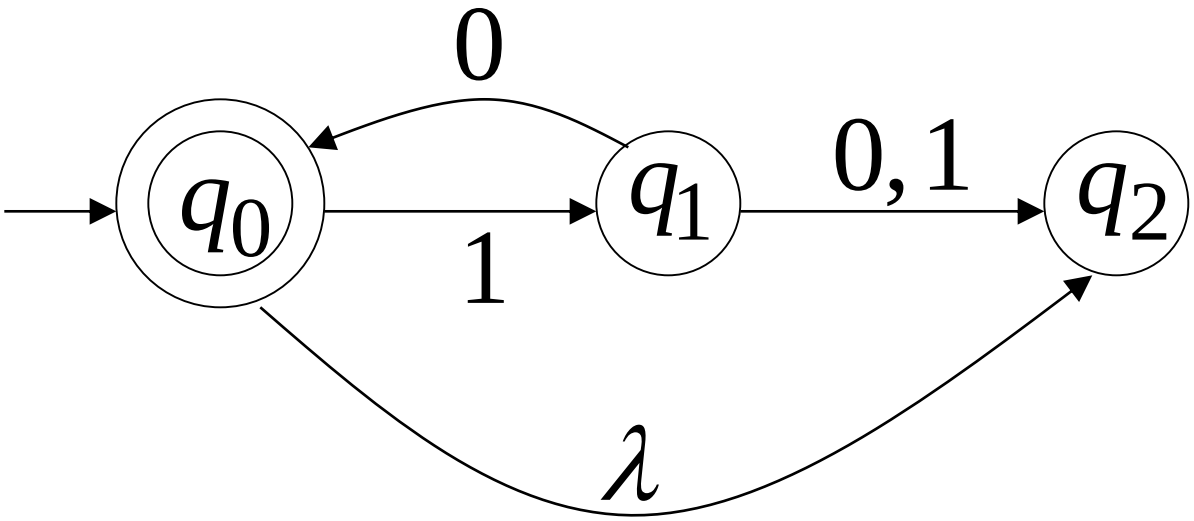
Another NFA Example



Language accepted

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

$$= \{10\}^*$$

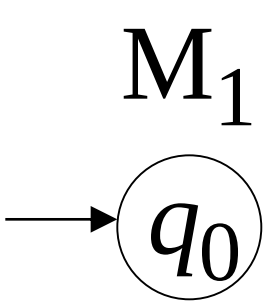


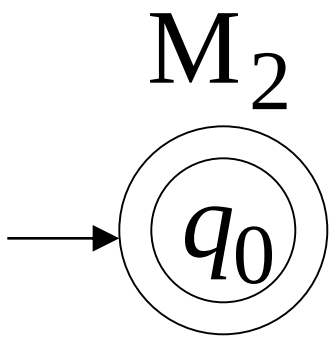
(redundant state)

Remarks:

- The λ symbol never appears on the input tape

- Simple automata:



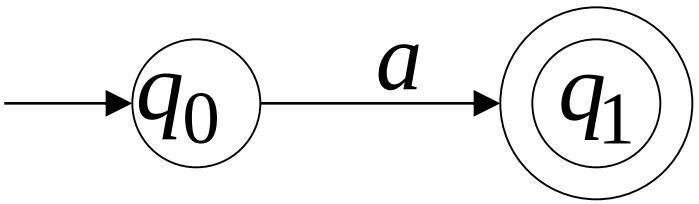


$L(M_1) = \{\}$

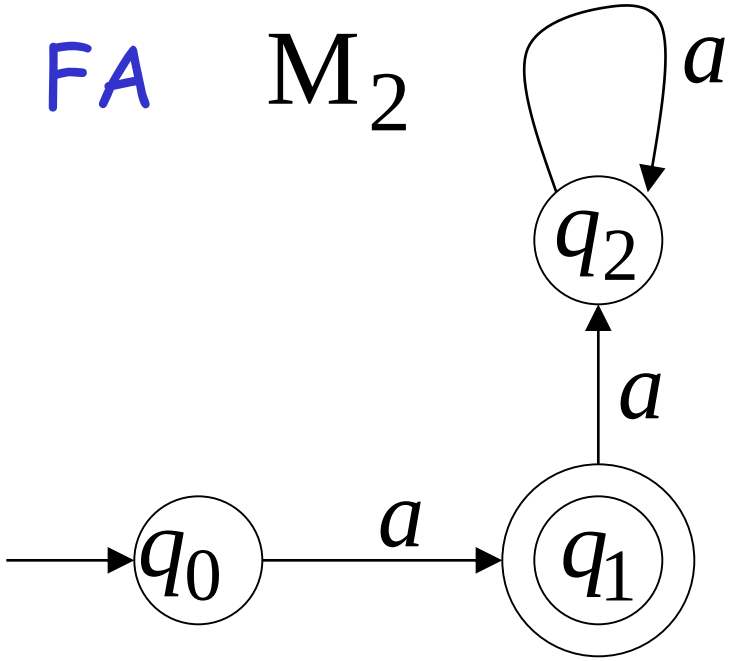
$L(M_2) = \{\lambda\}$

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

NFA M_1



FA 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. $\{q_0, q_1, q_2\}$

Σ: Input aplhabet, i.e. $\{a, b\}$

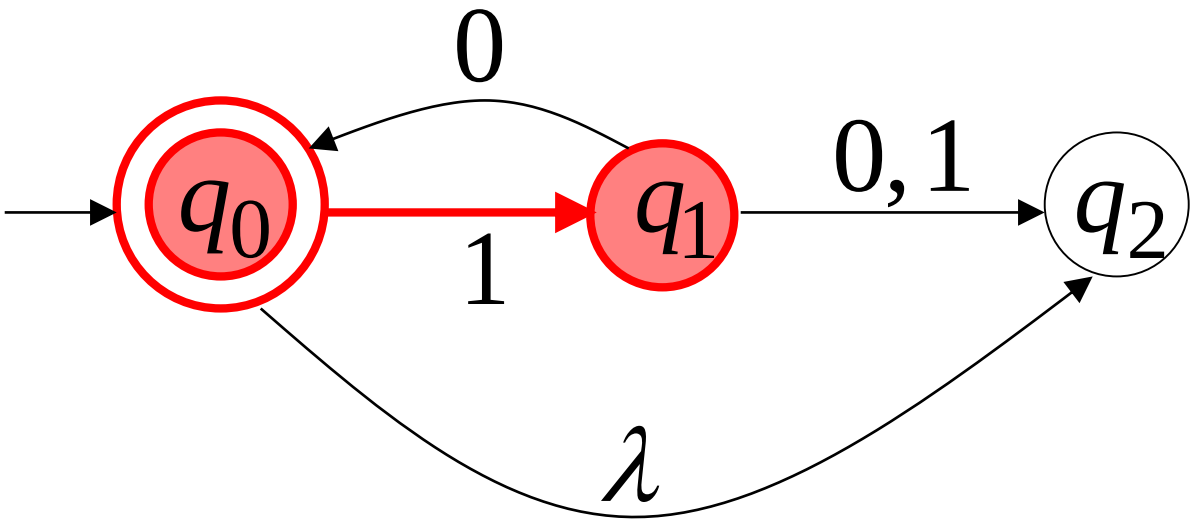
δ: Transition function

q_0: Initial state

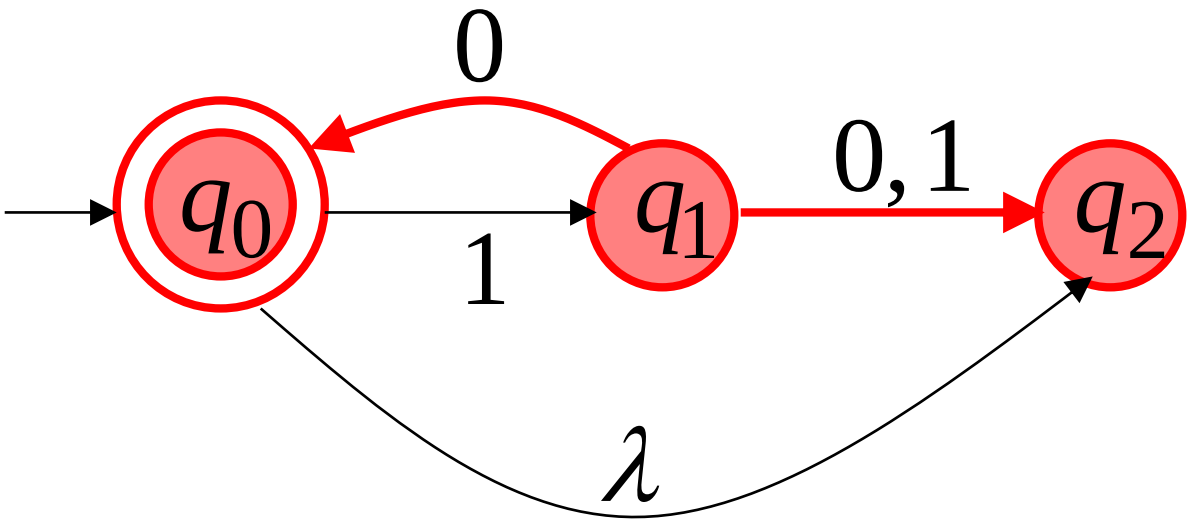
F: Accepting states

Transition Function δ

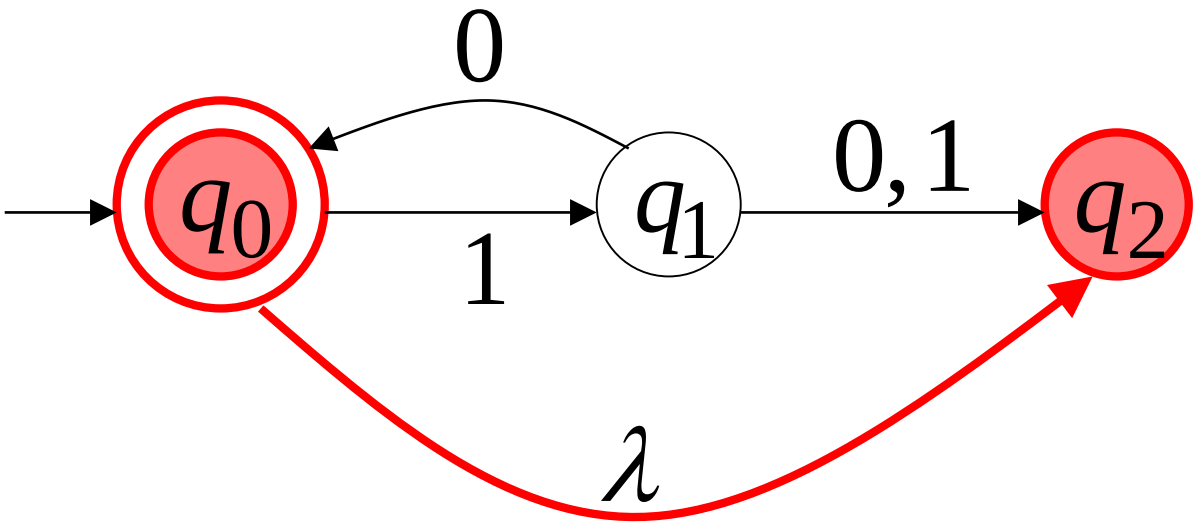
$$\delta(q_0, 1) = \{q_1\}$$



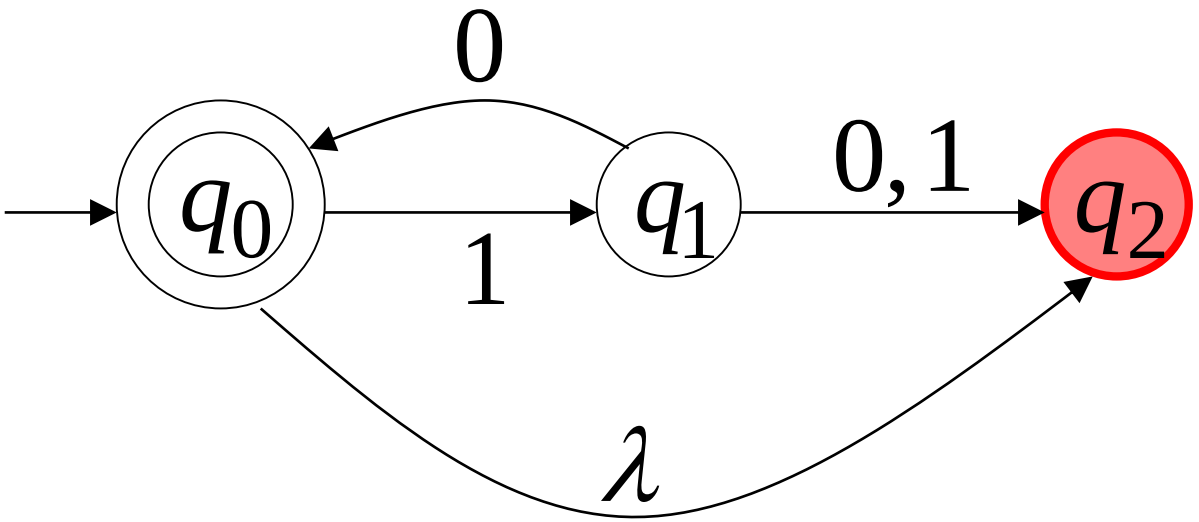
$$\delta(q_1,0)=\{q_0,q_2\}$$



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

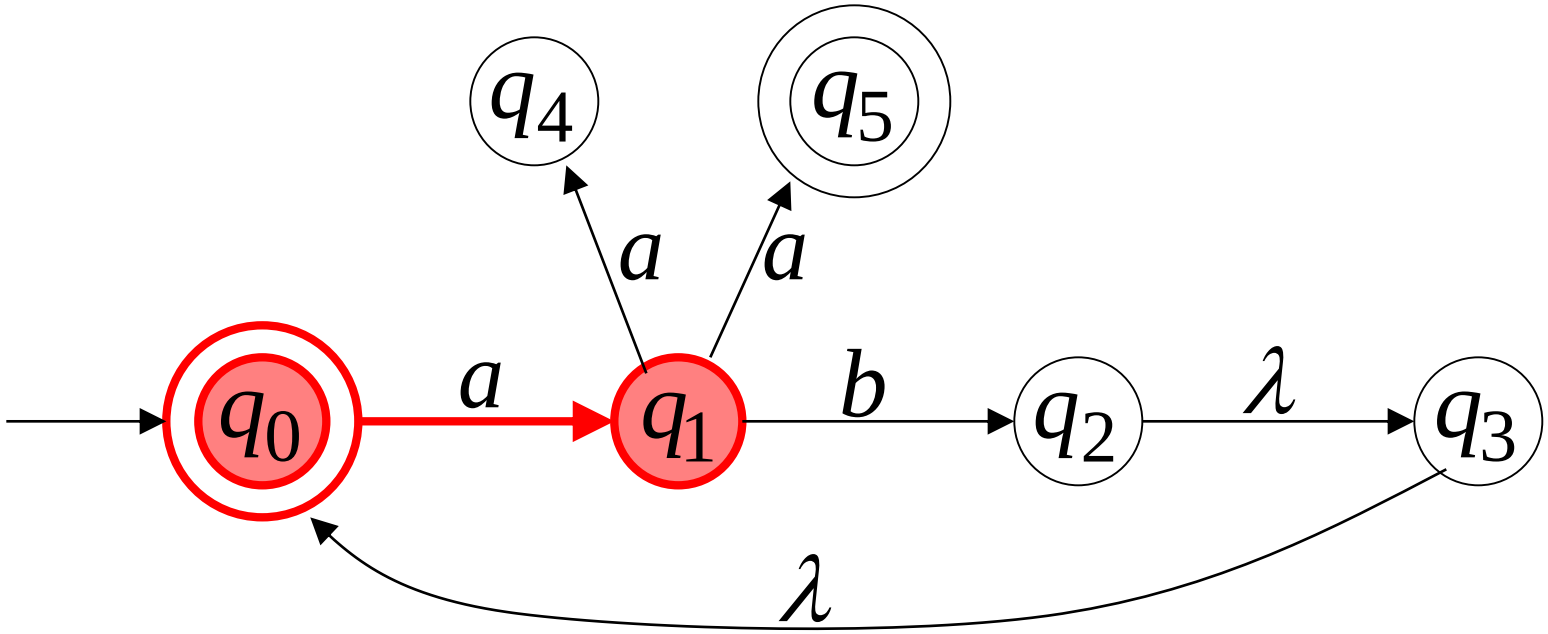


$$\delta(q_2,1)=\varnothing$$

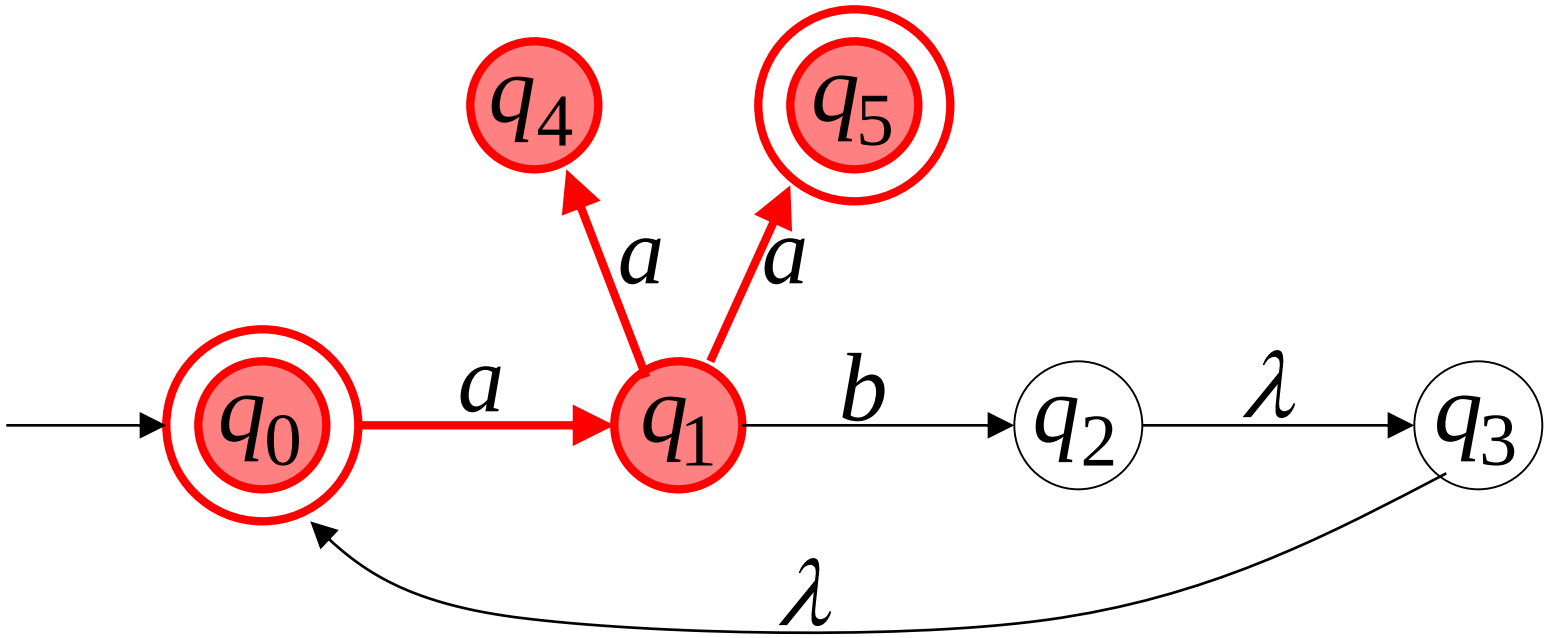


Extended Transition Function δ^*

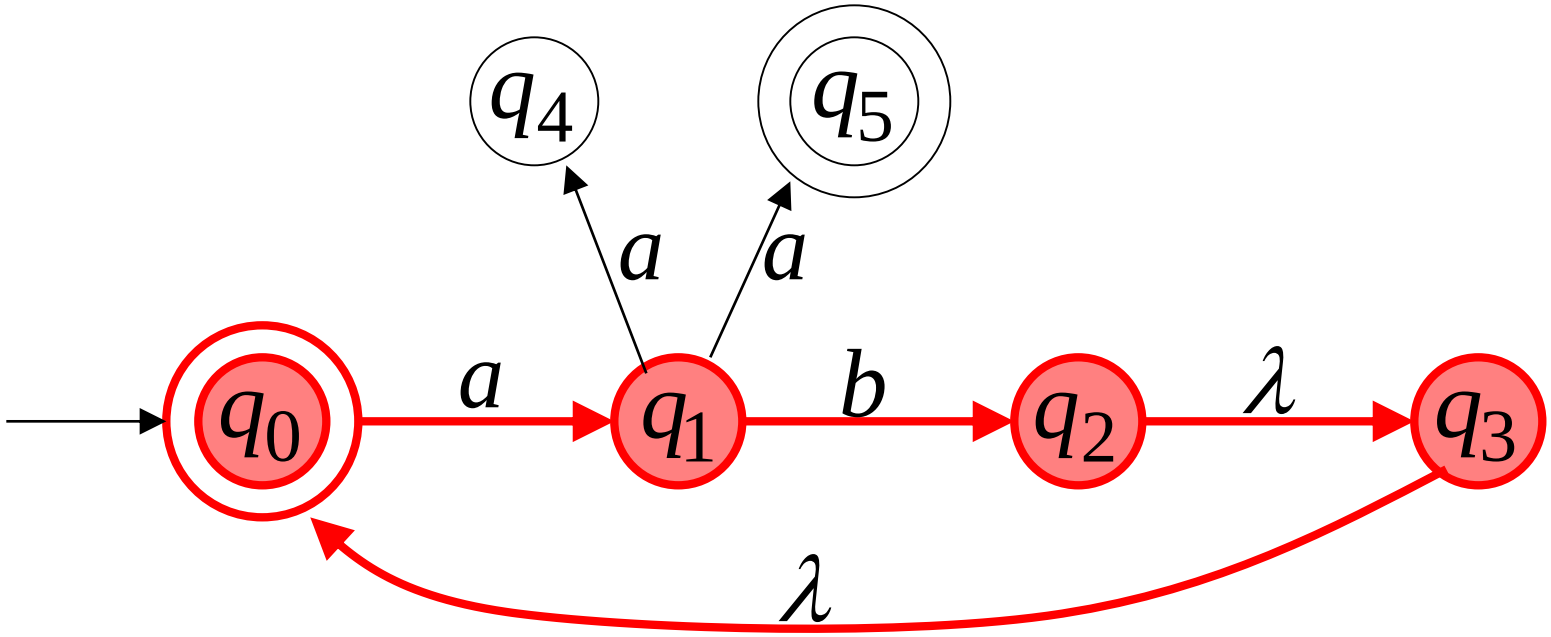
$$\delta^*(q_0, a) = \{q_1\}$$



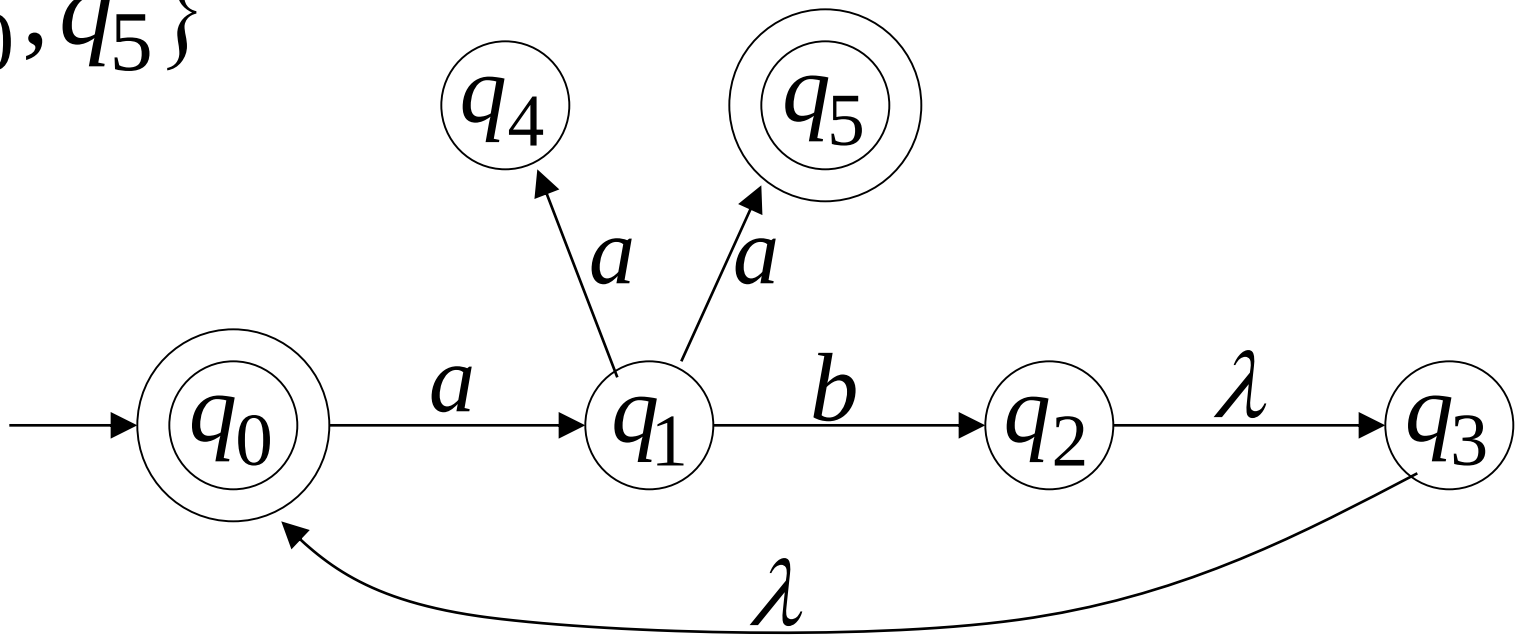
$$\delta^*(q_0, aa) = \{q_4, q_5\}$$



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



$F = \{q_0, q_5\}$



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

$\searrow \in F$

$abaa \in L(M)$

Formally

The language accepted by NFA M is:

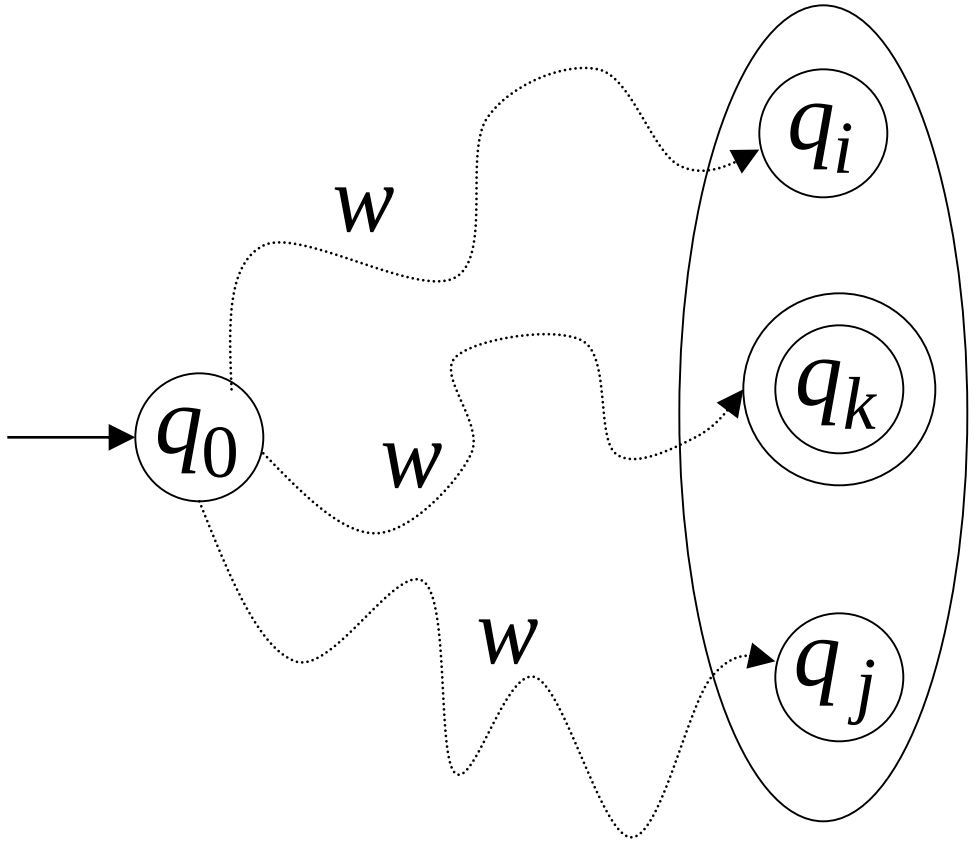
$$L(M) = \{w_1, w_2, w_3, \ldots\}$$

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

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

$w \in L(M)$

$\delta^*(q_0, w)$



$q_k \in F$