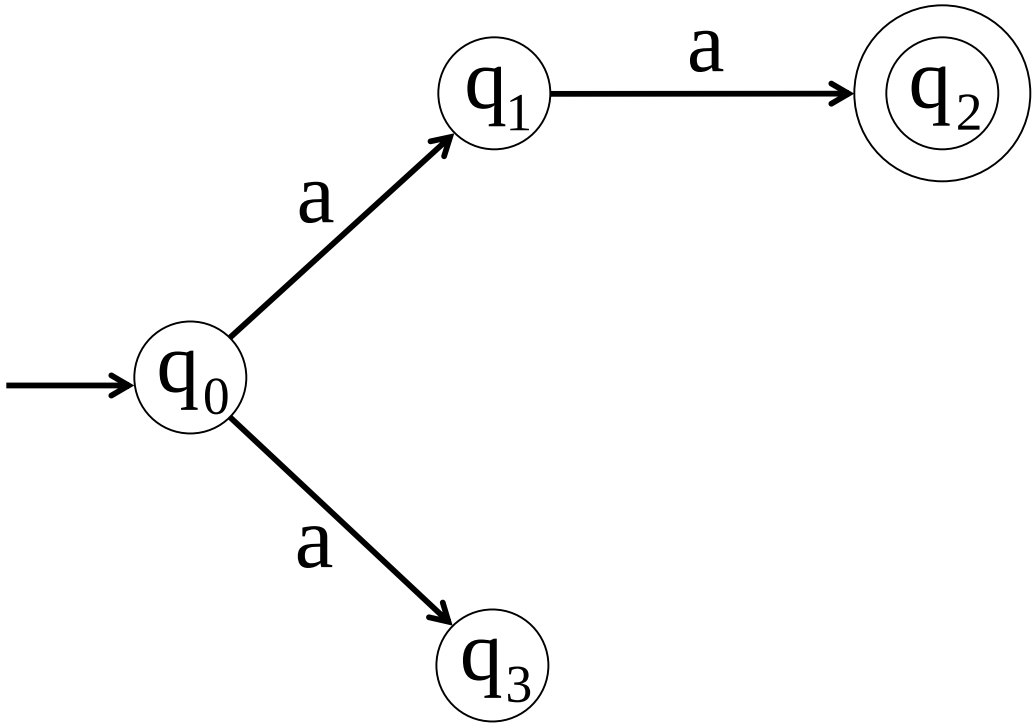
Nondeterministic Finite Automata

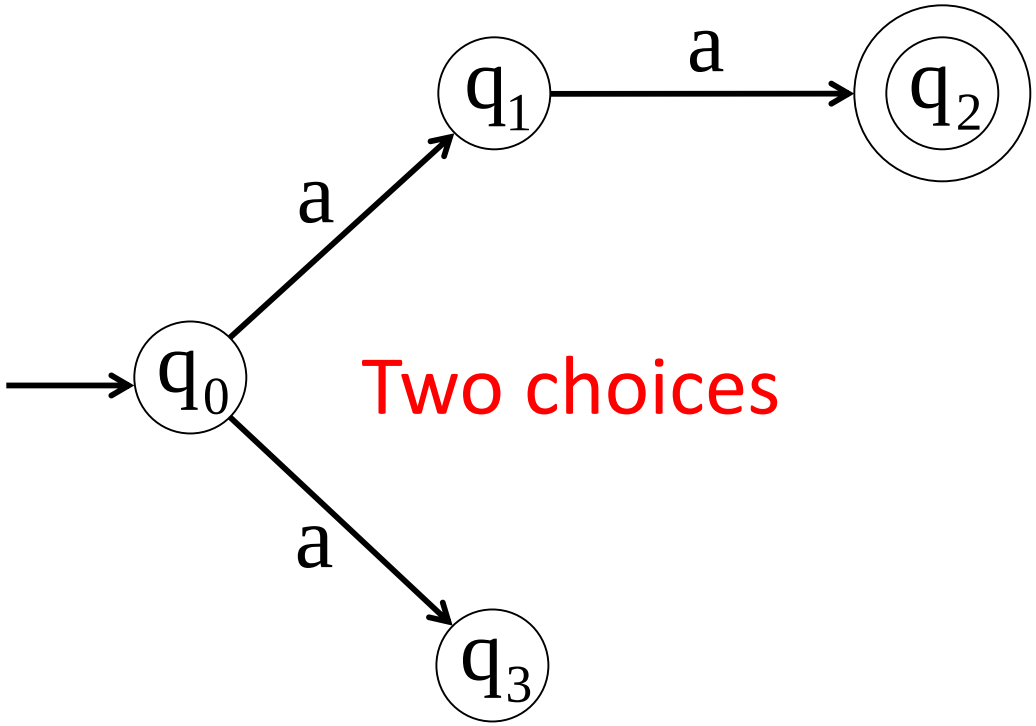
Nondeterministic Finite Accepter (NFA)

- Alphabet = {a}



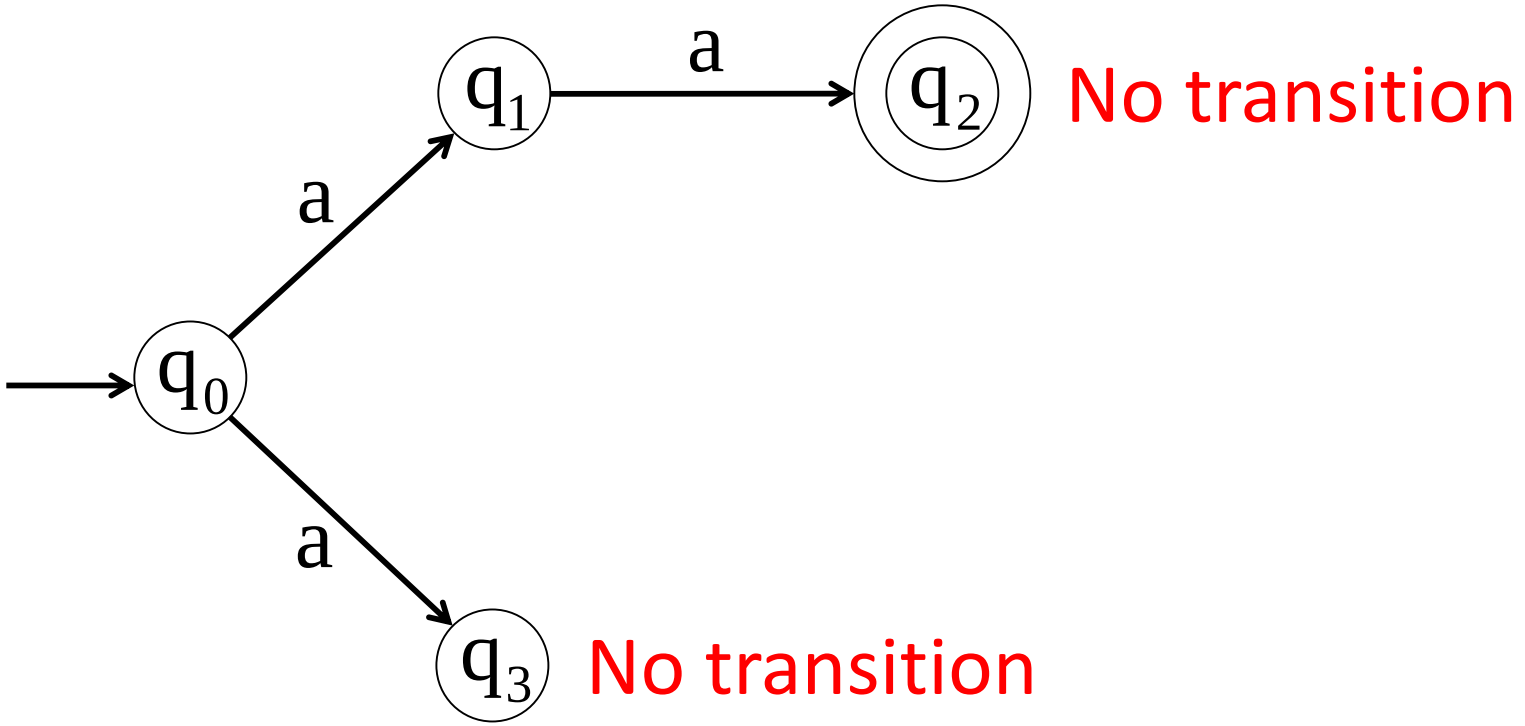
Nondeterministic Finite Accepter (NFA)

- Alphabet = {a}

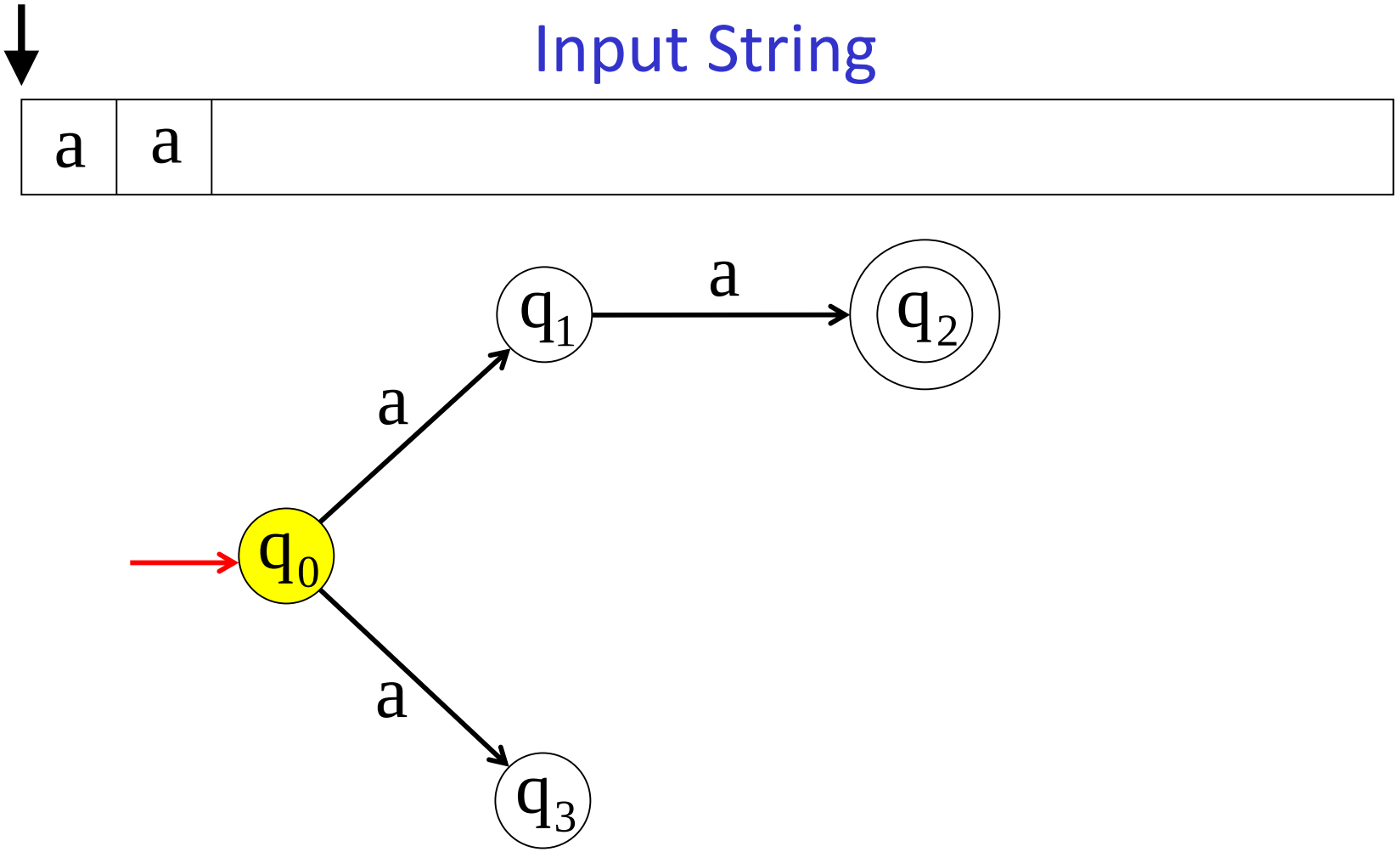


Nondeterministic Finite Accepter (NFA)

- Alphabet = {a}

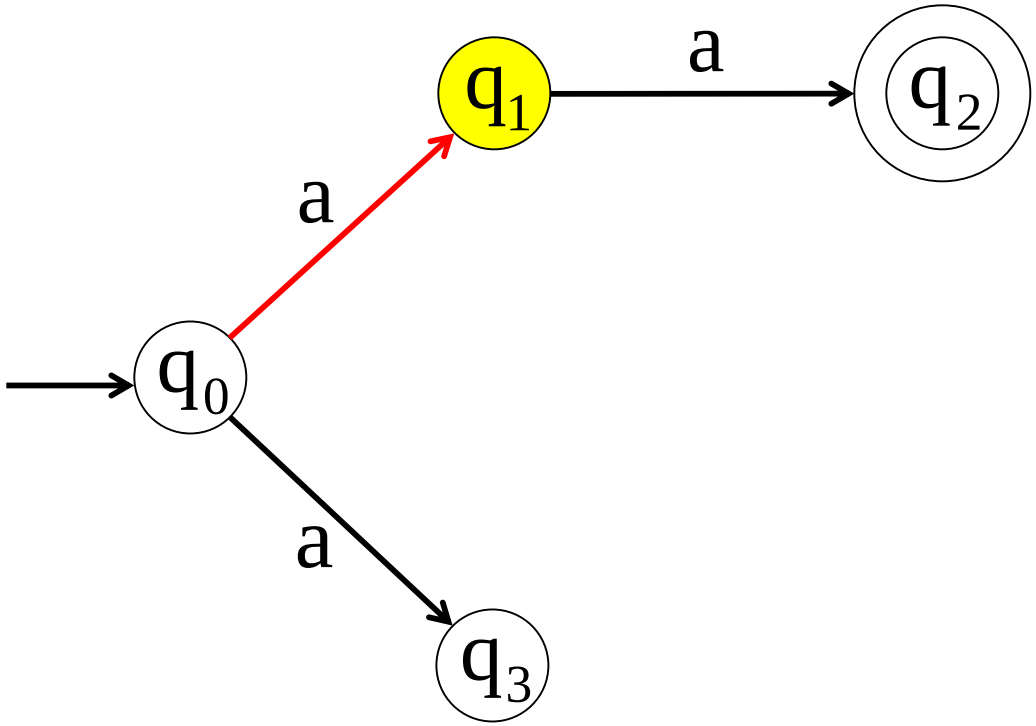


Nondeterministic Finite Accepter (NFA)

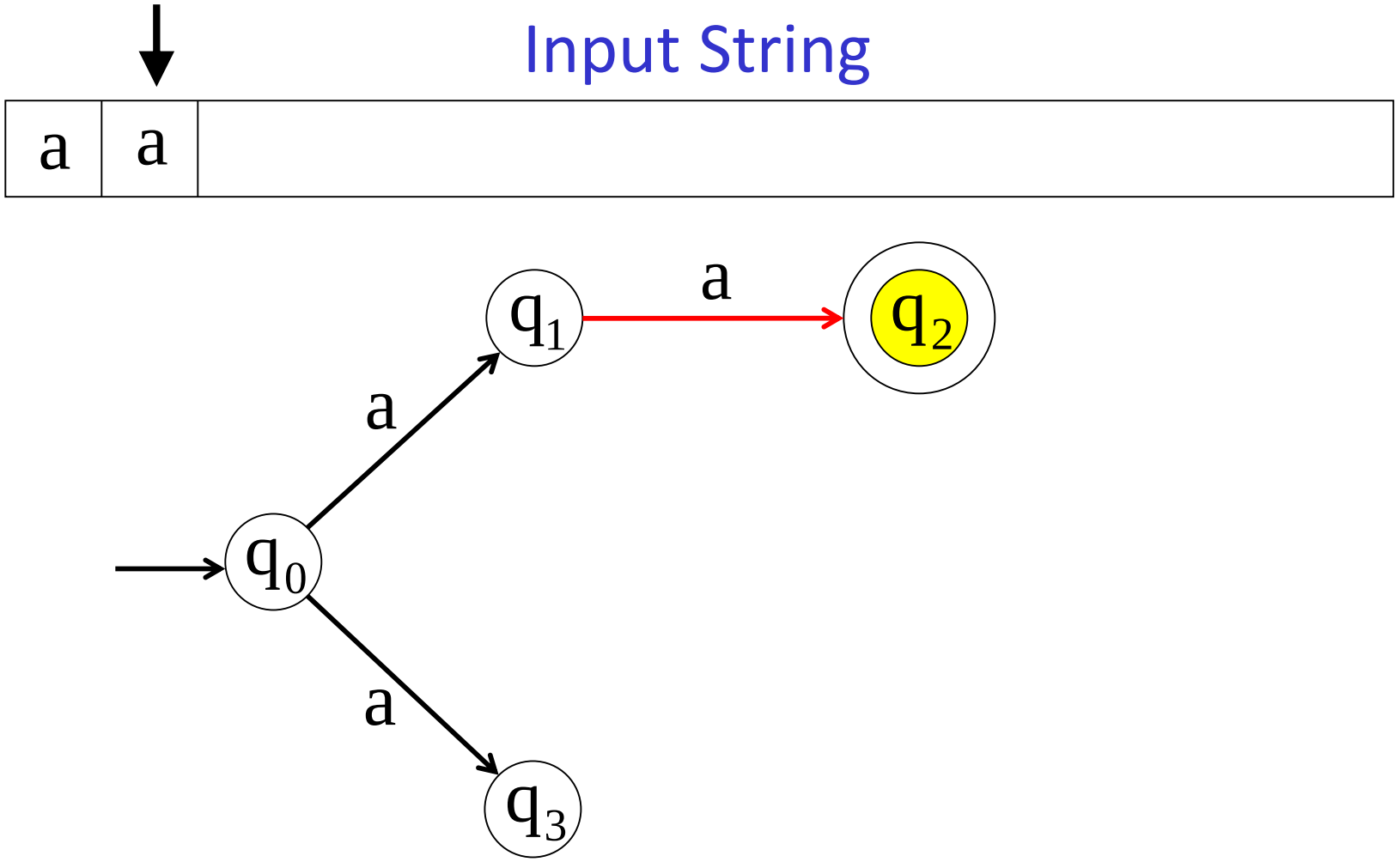


First Choice

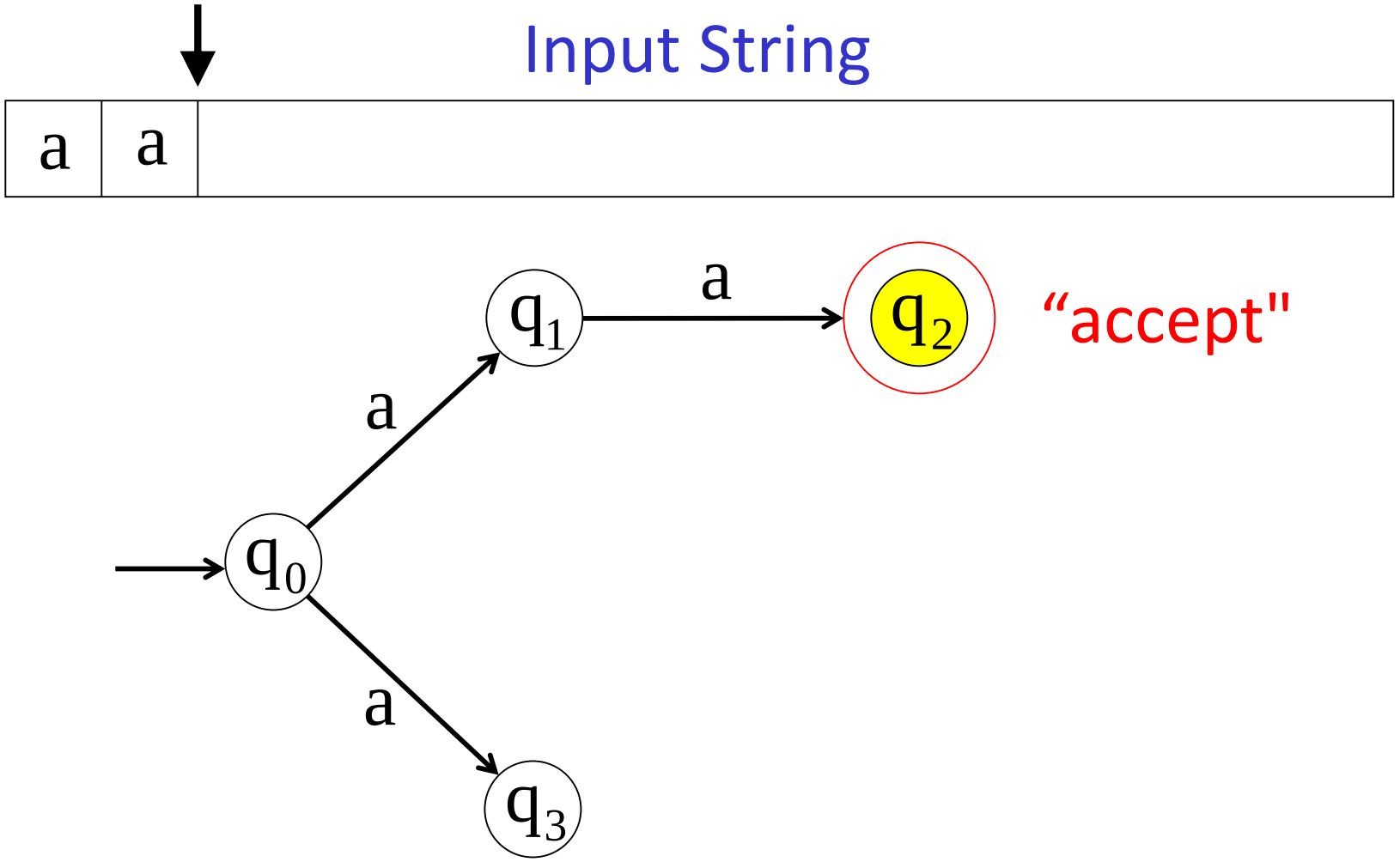
Input String



First Choice



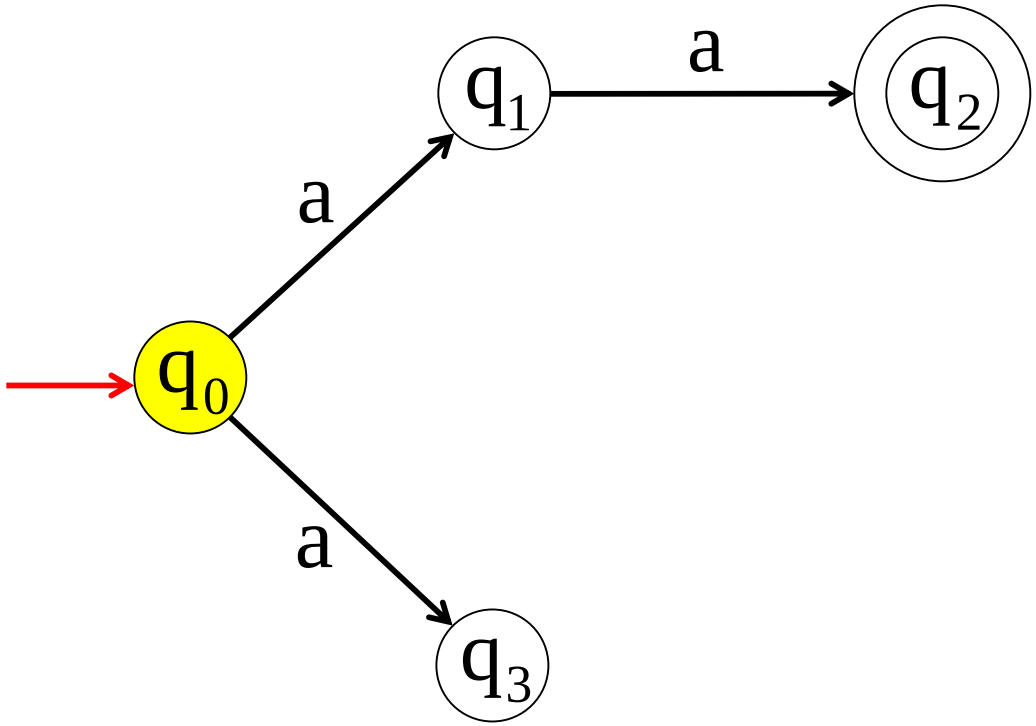
First Choice



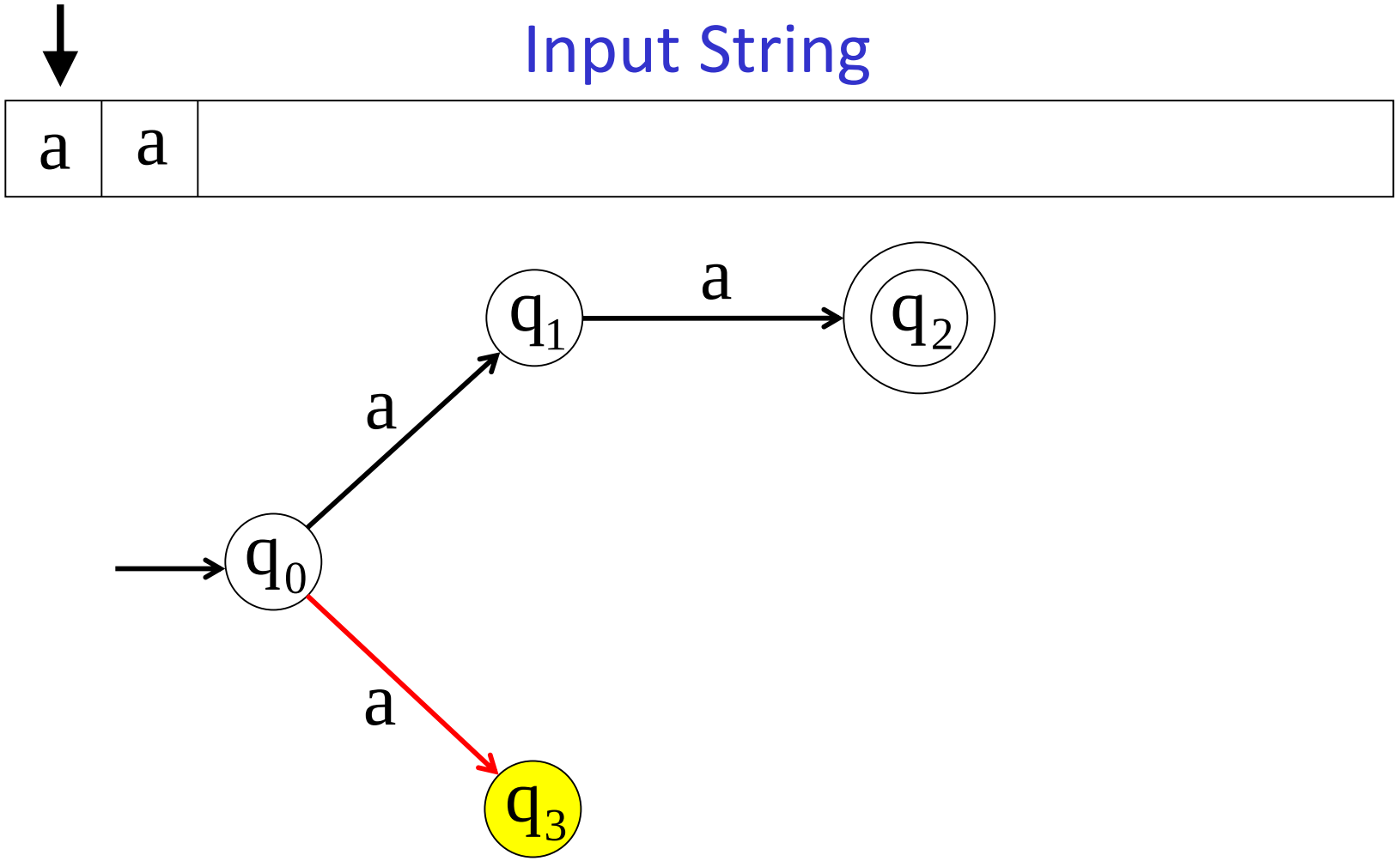
All input is consumed

Second Choice

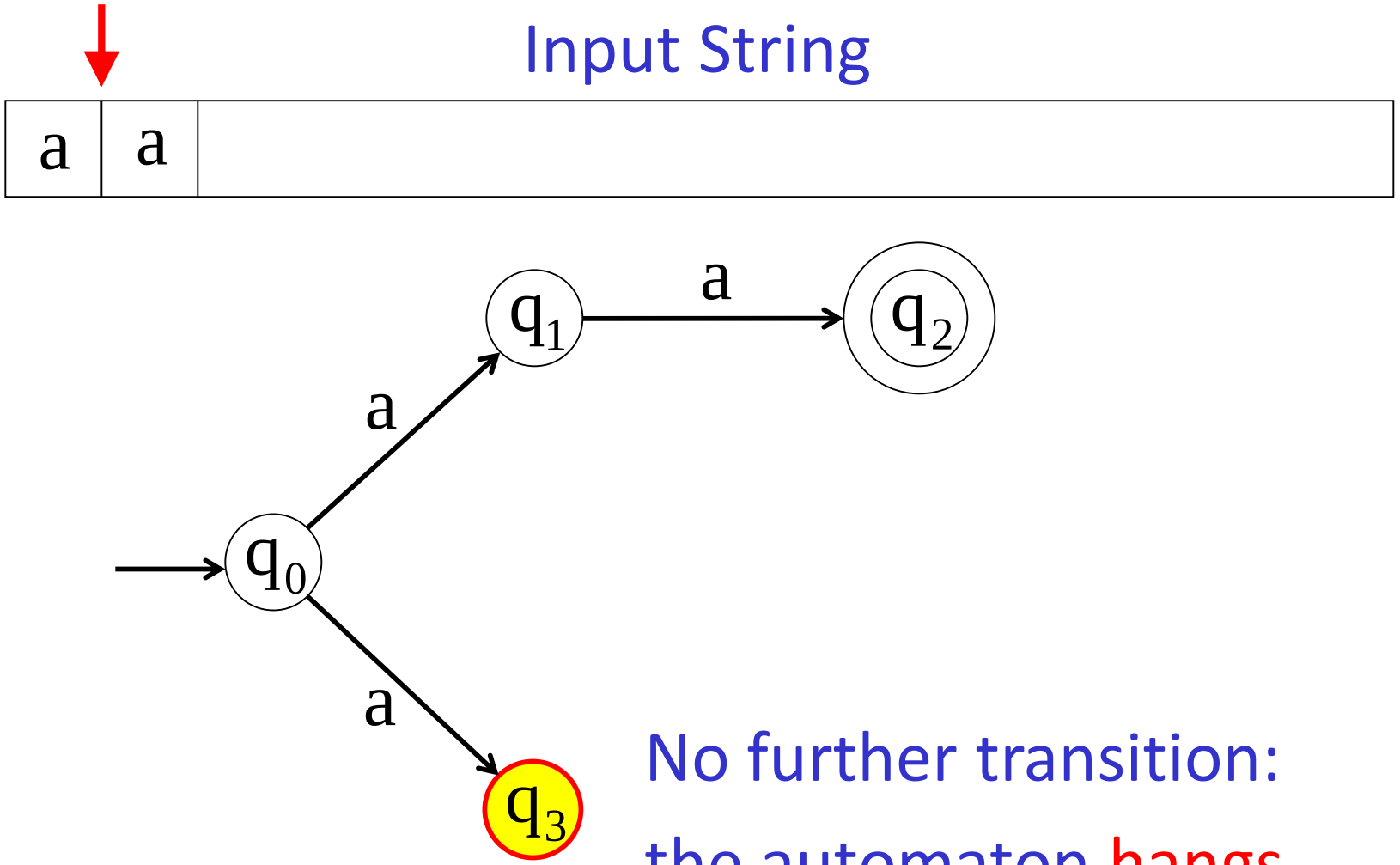
Input String



Second Choice



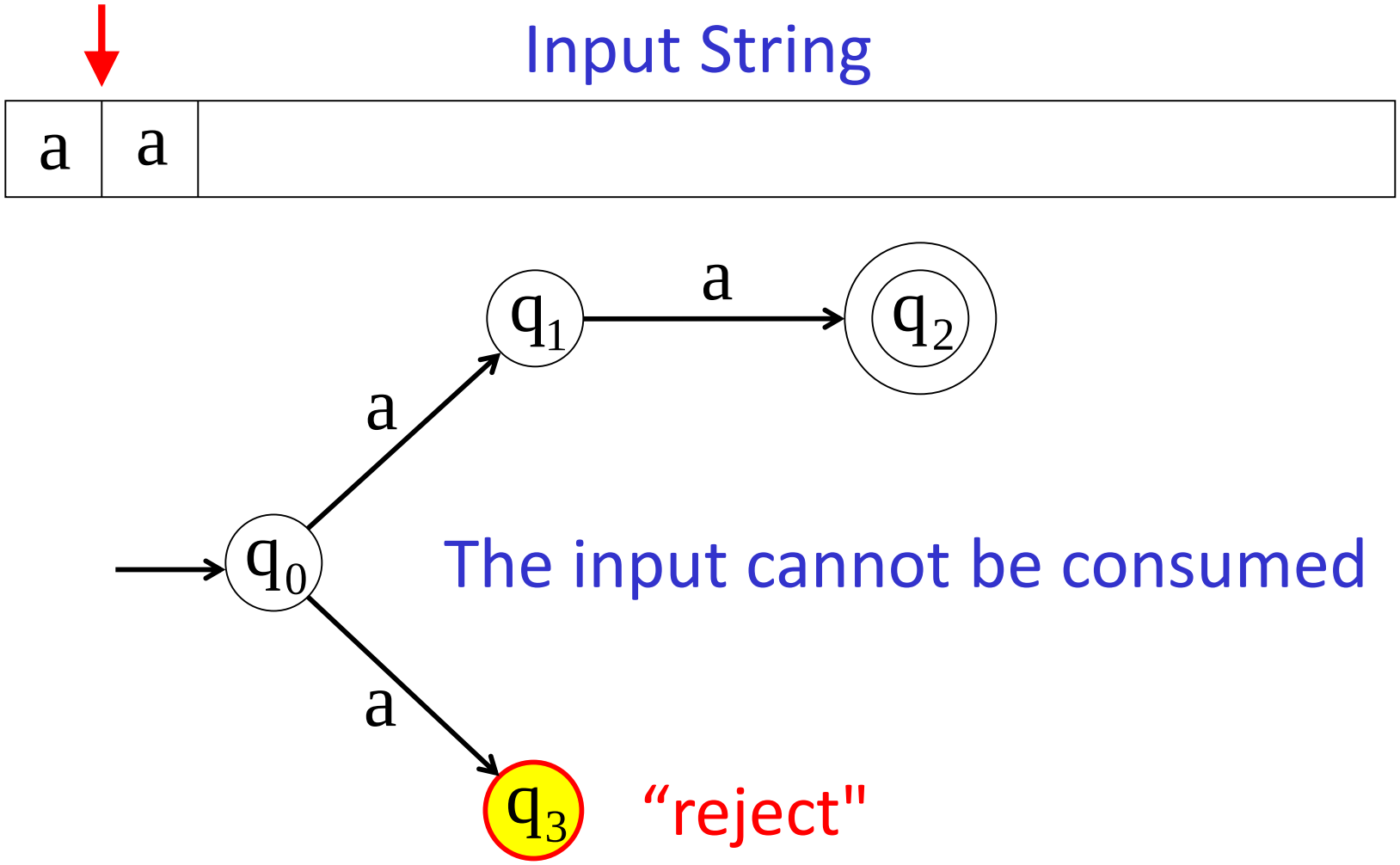
Second Choice



No further transition:

the automaton hangs

Second Choice



NFA: Acceptance

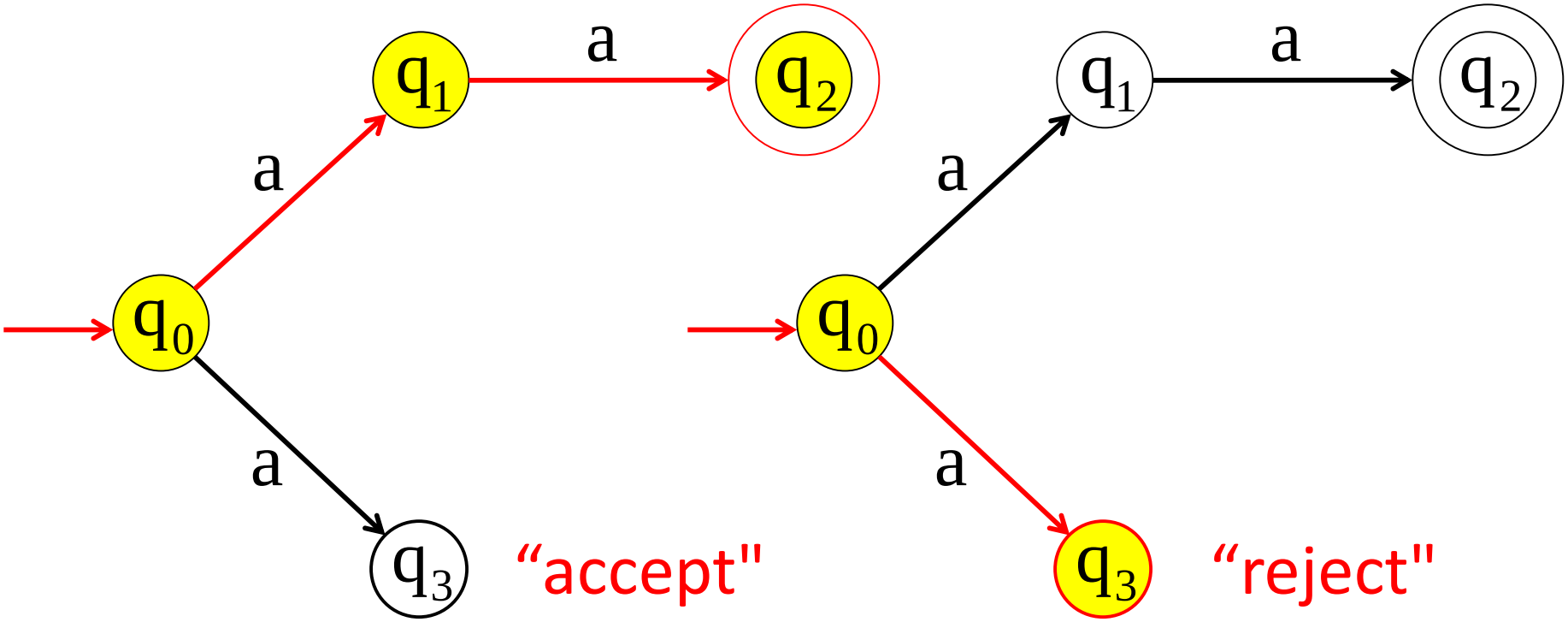
- An NFA **accepts** a string:

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

  - all the input is **consumed** and
  - the automaton is in **a final state**.

Example

aa is accepted by the NFA



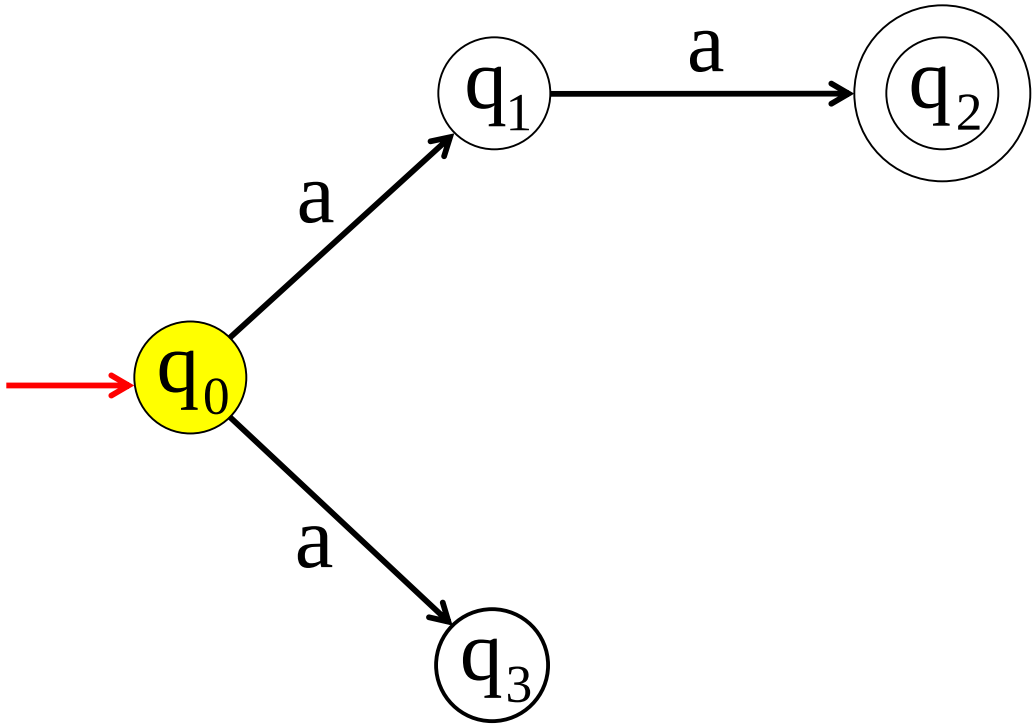
"accept"

"reject"

because this computation accepts

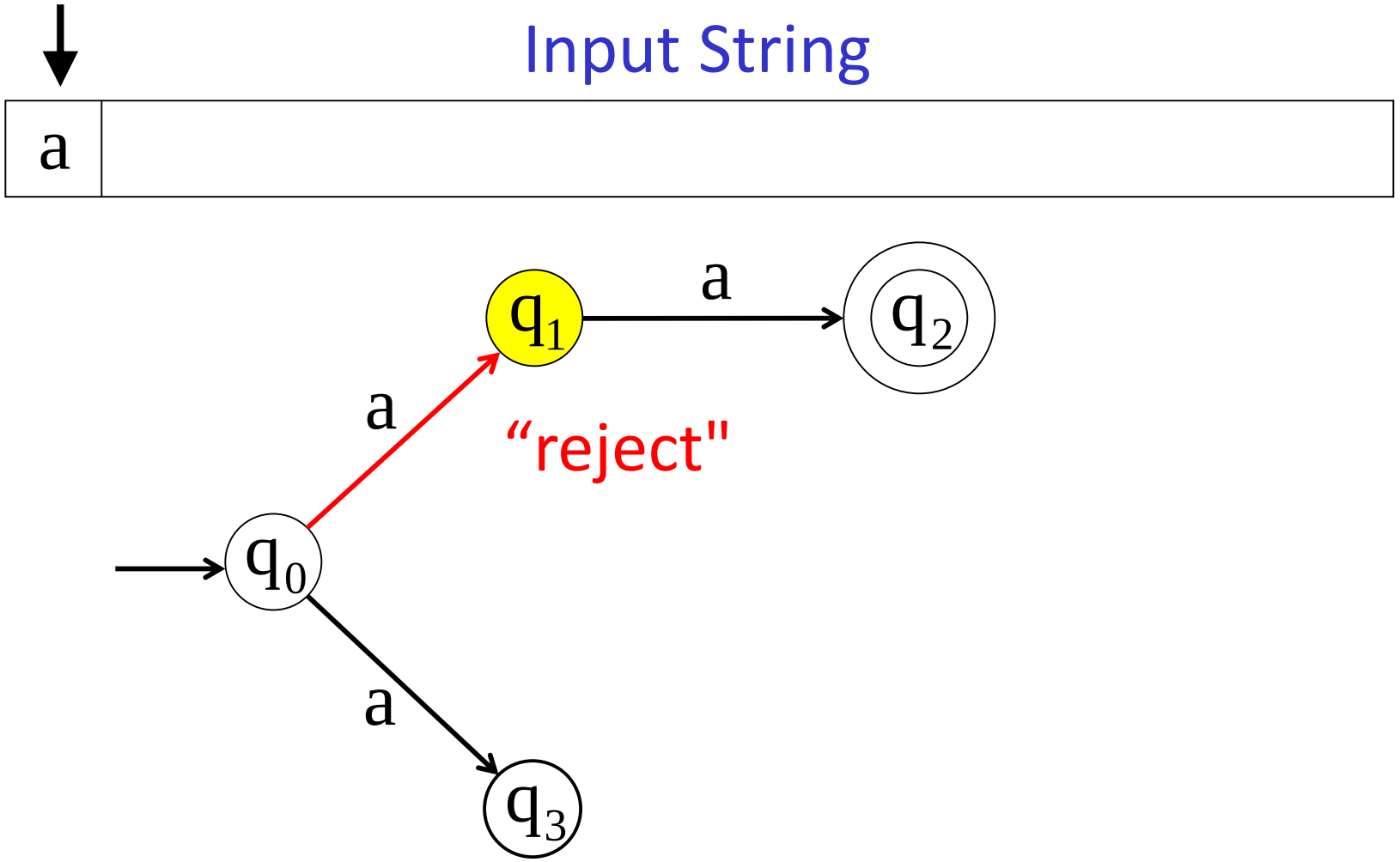
Rejection Example

Input String



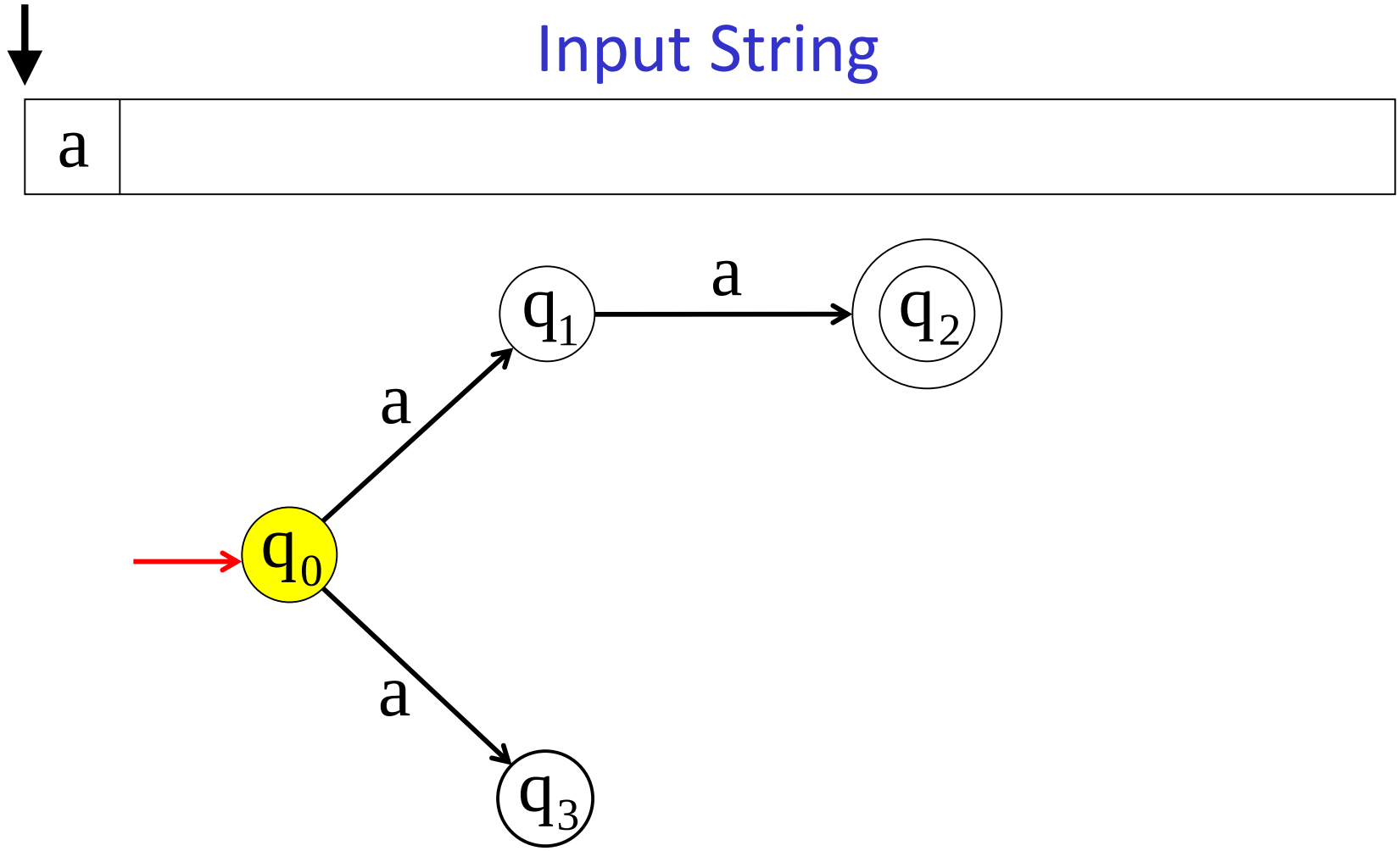
First Choice

Input String

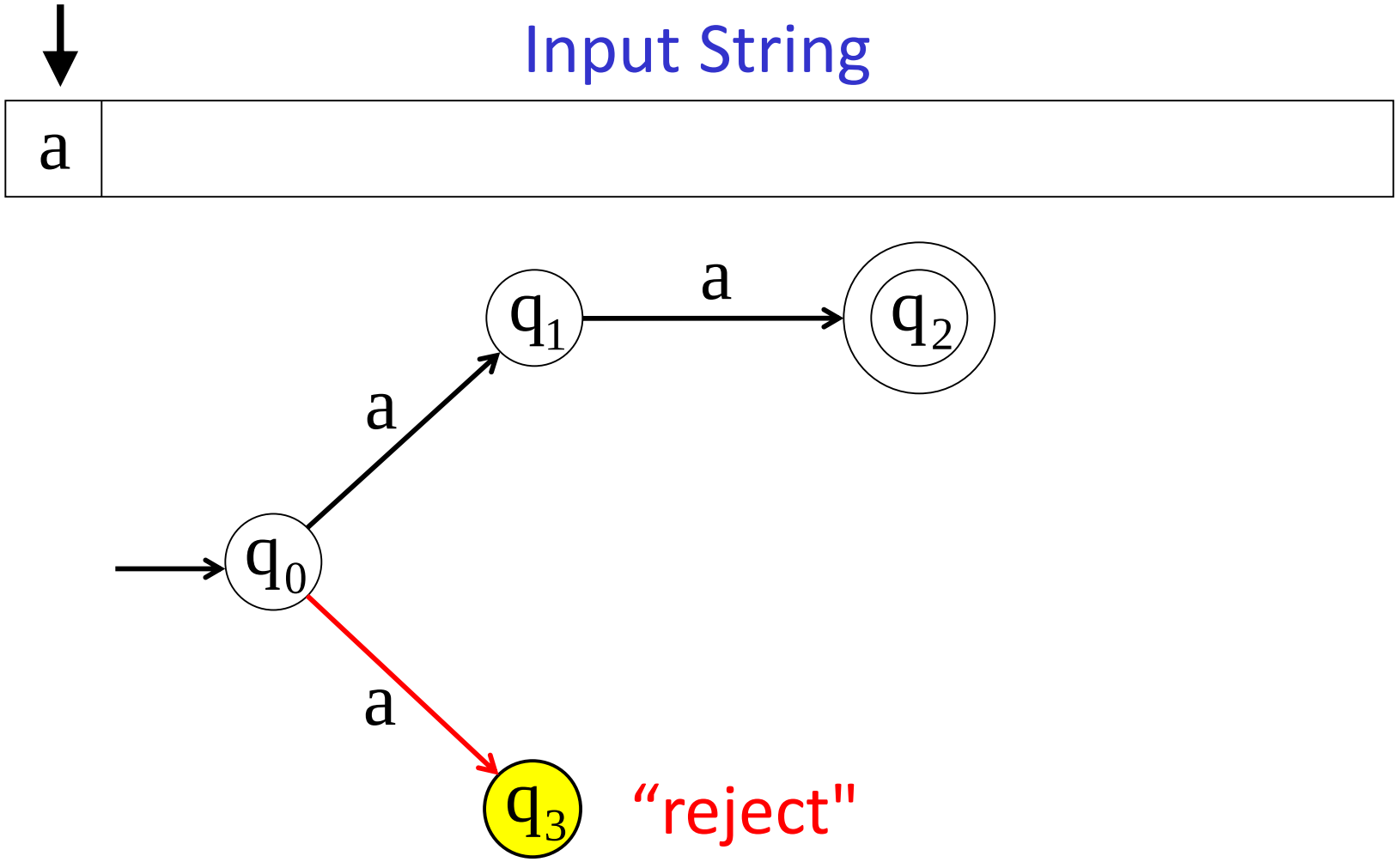


Second Choice

Input String



Second Choice



NFA: Rejection

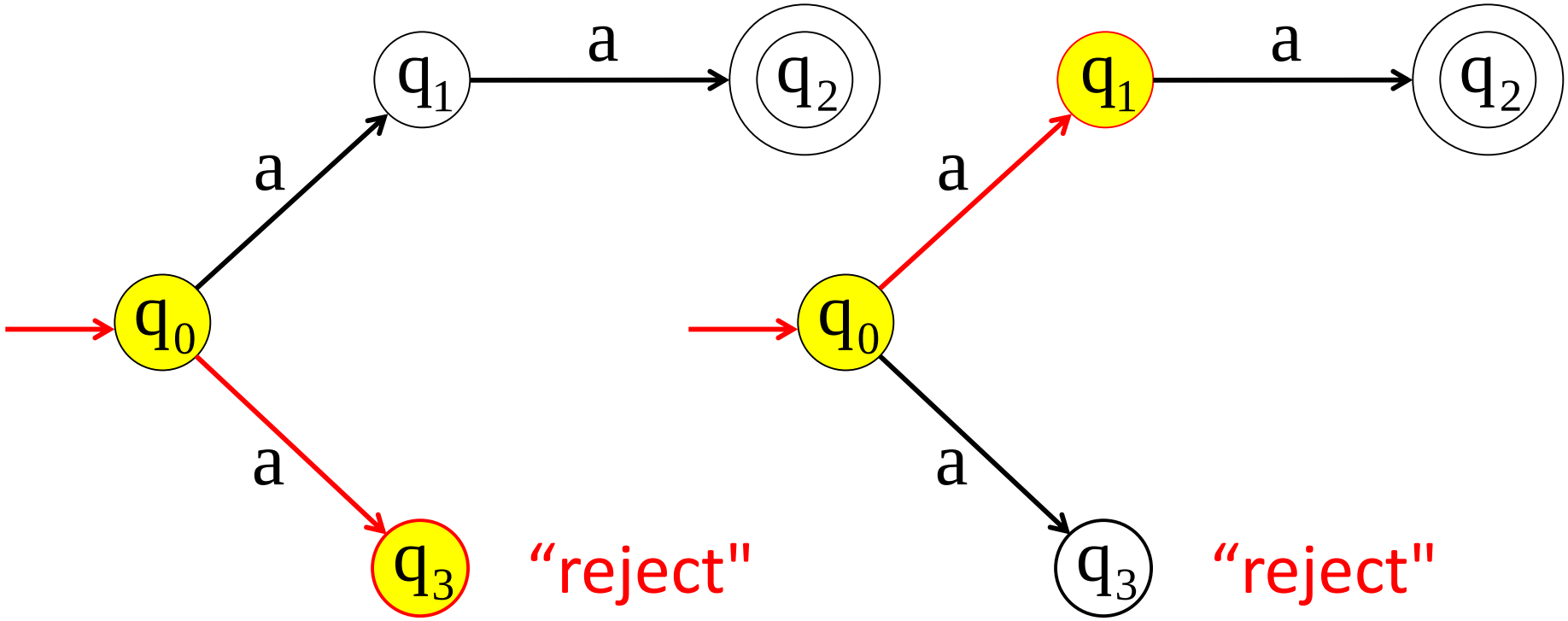
An NFA **rejects** a string:

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 can**not be consumed**

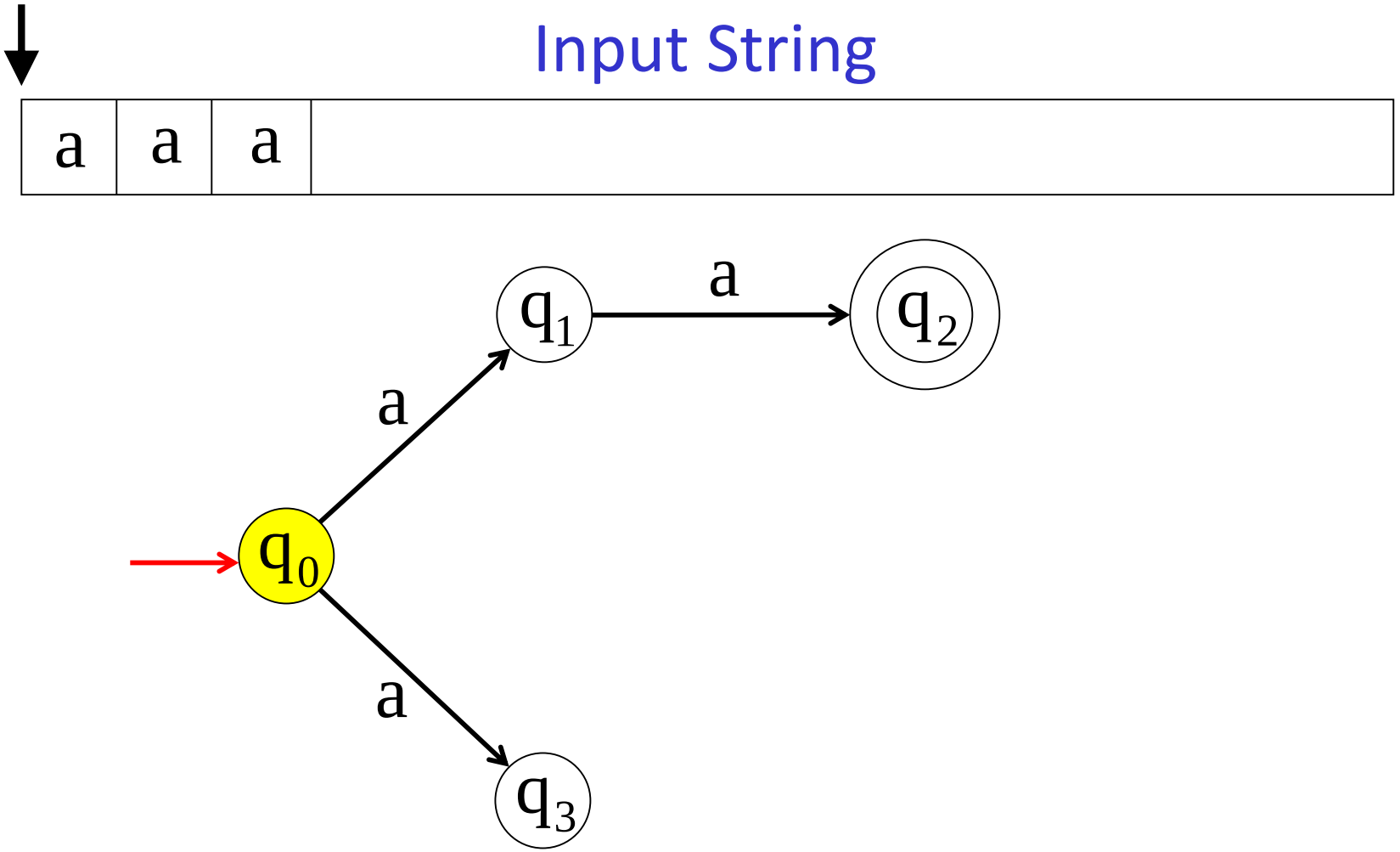
Example

a is rejected by the NFA



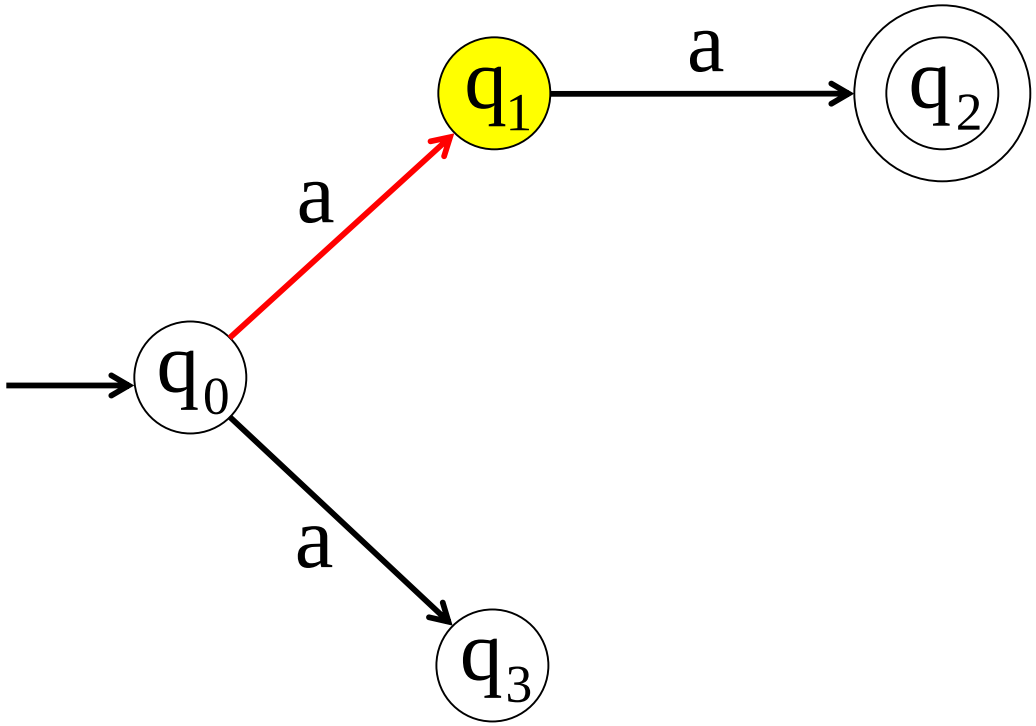
All **possible** computations leads to rejection

Rejection Example

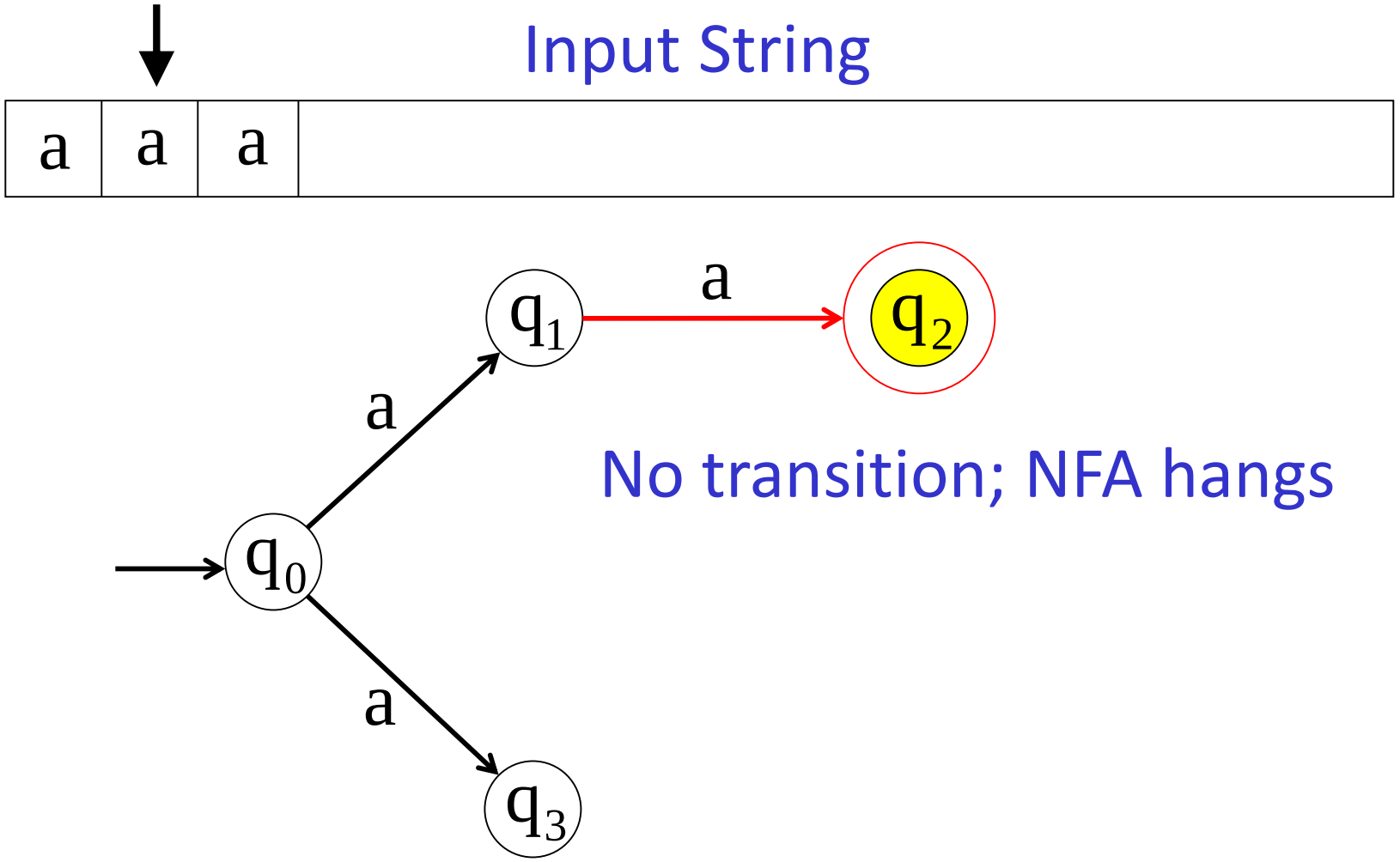


First Choice

Input String

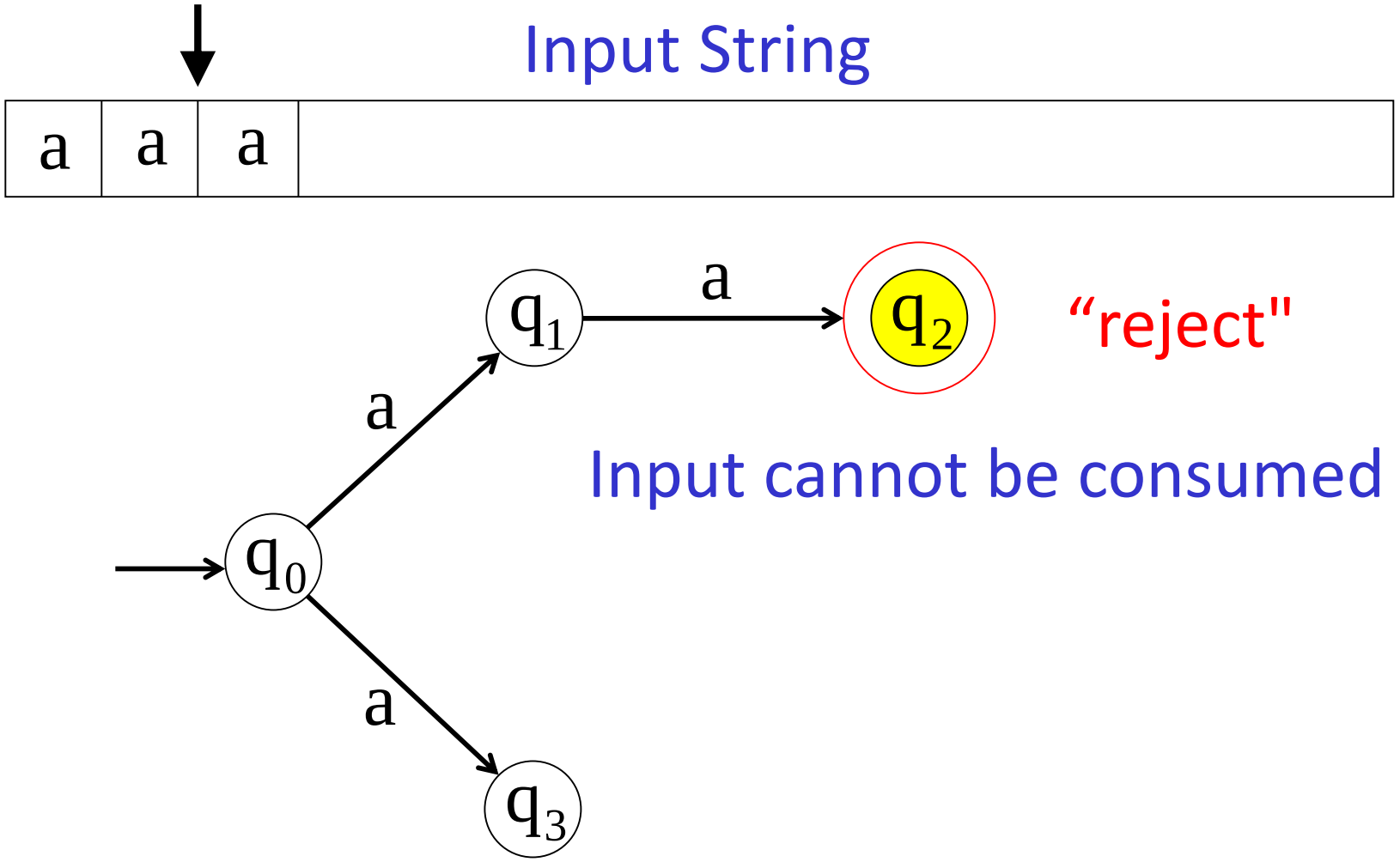


First Choice

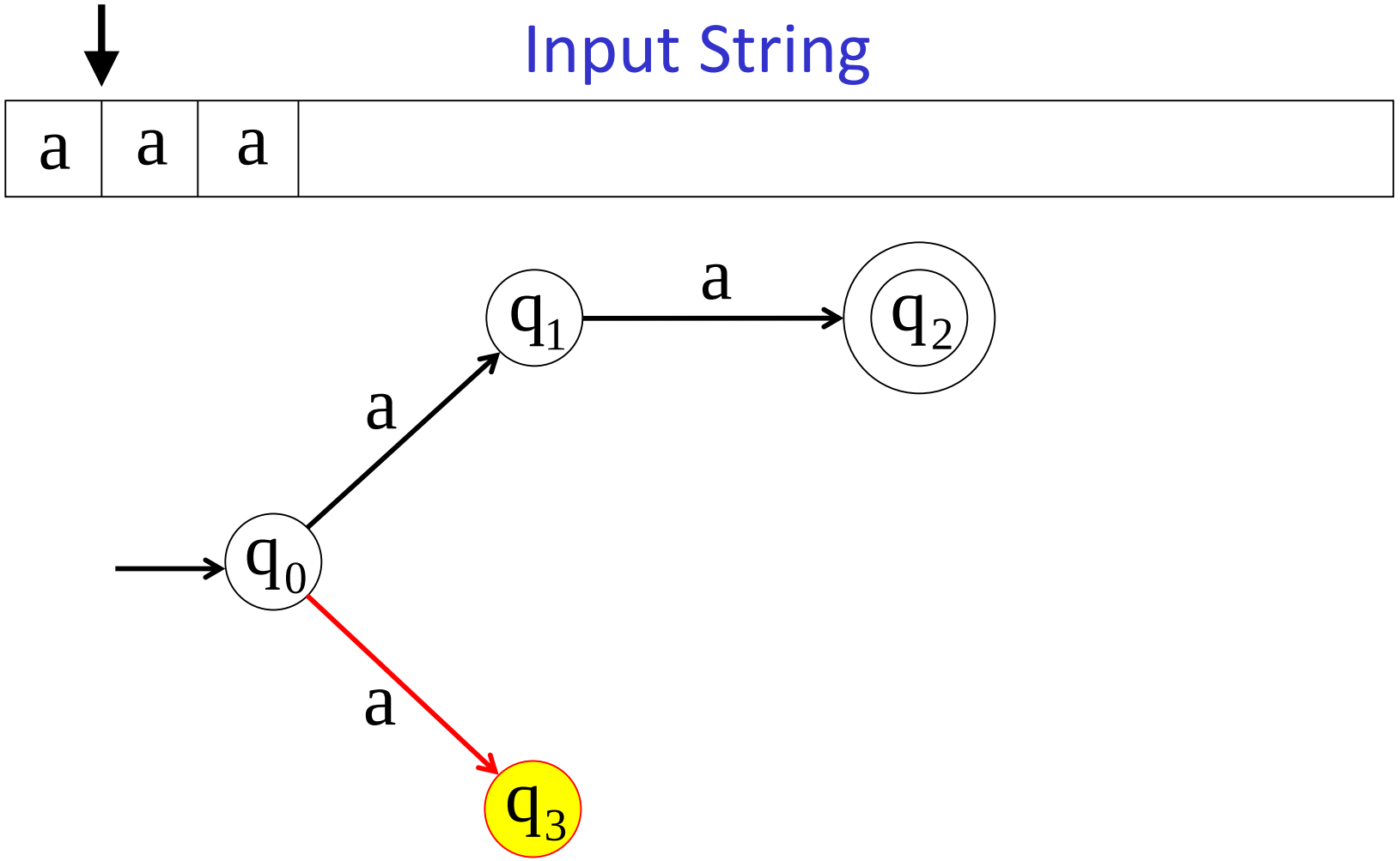


No transition; NFA hangs

First Choice

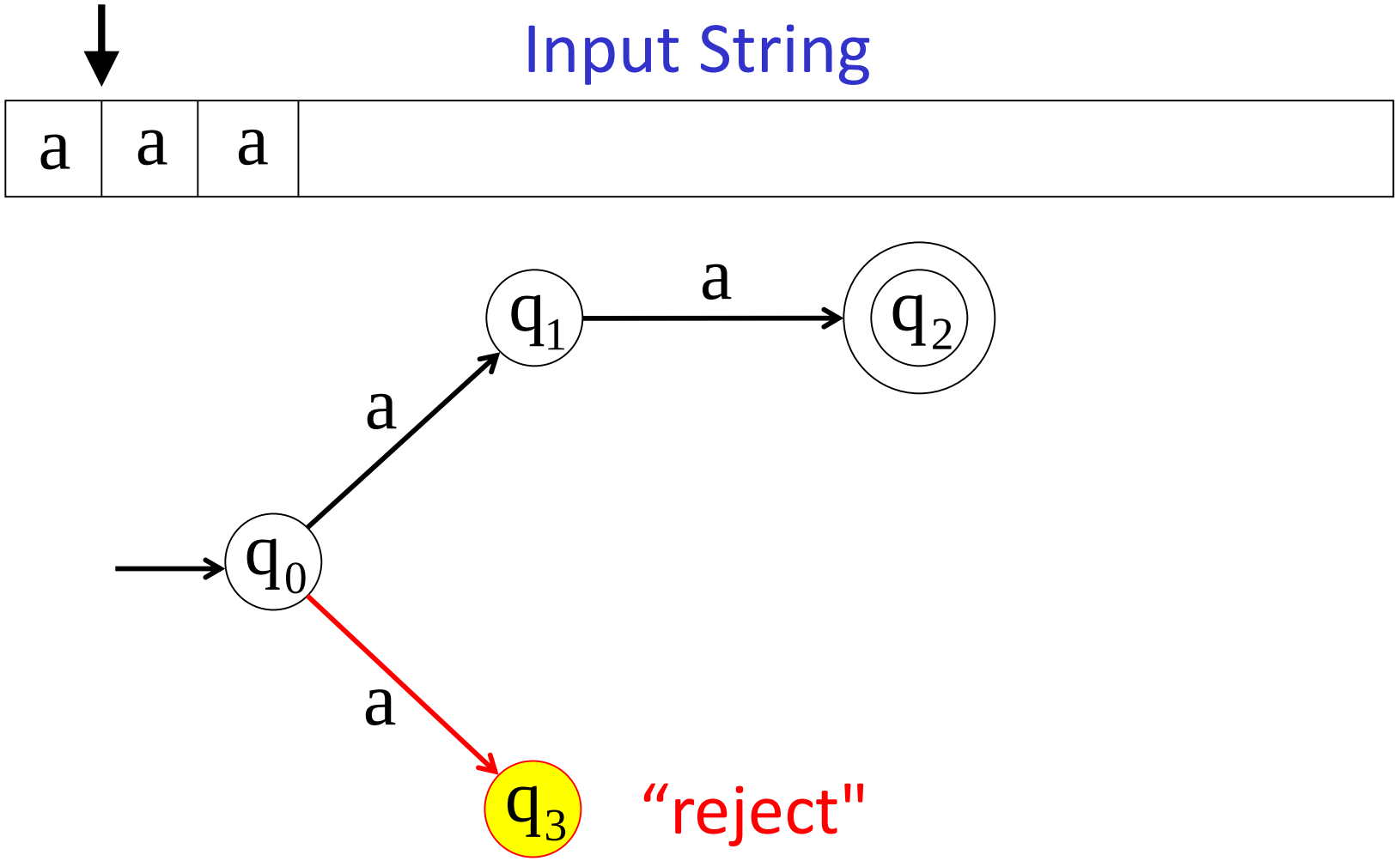


Second Choice



No transition; NFA hangs

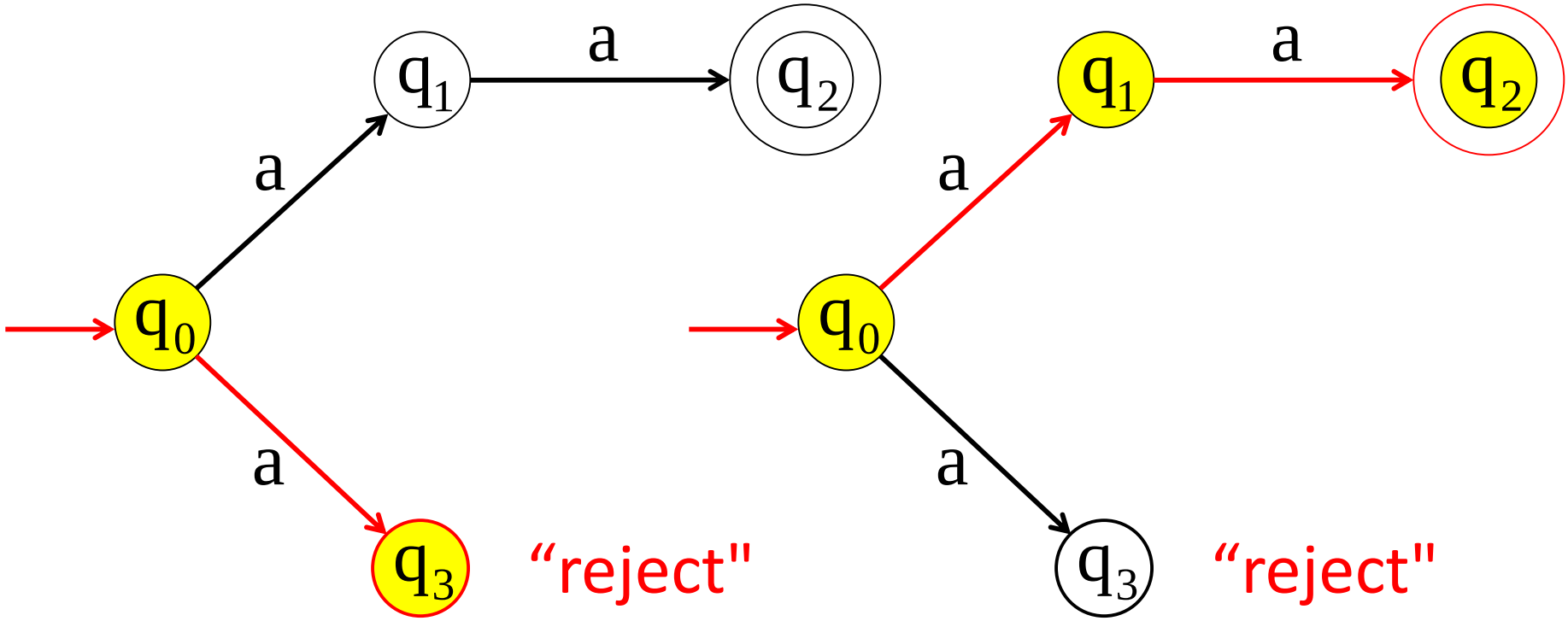
Second Choice



Input cannot be consumed

Example

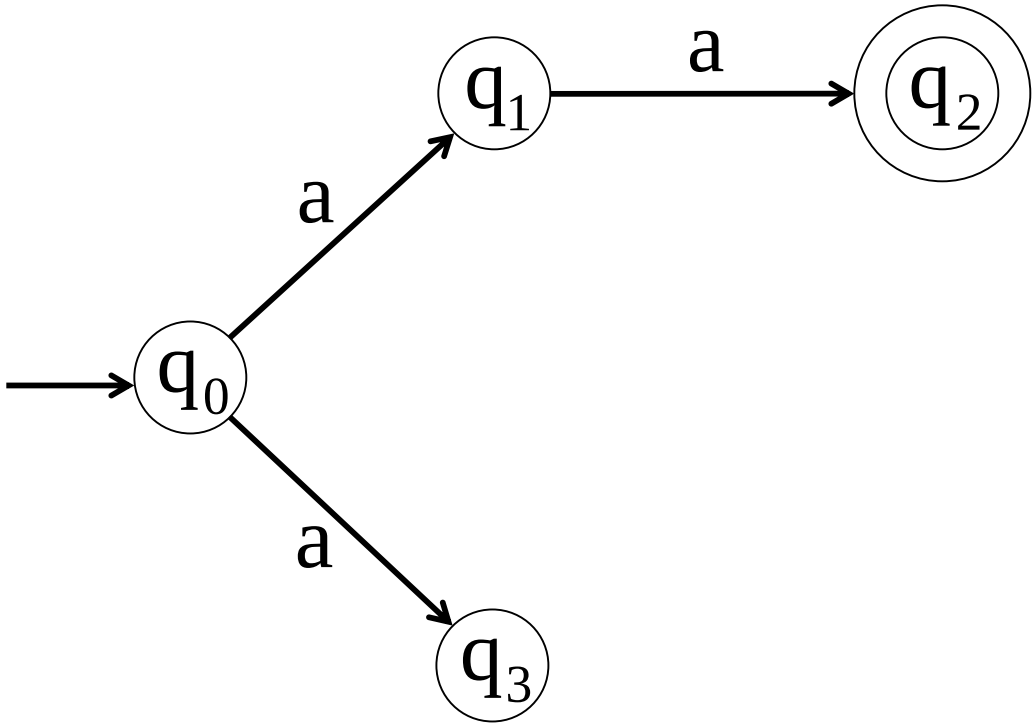
aaa is rejected by the NFA



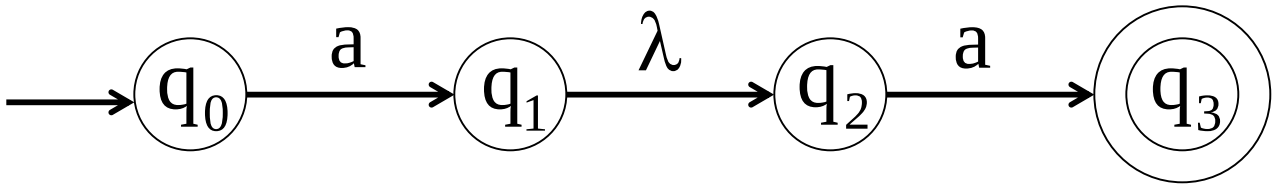
All **possible** computations leads to rejection

Language

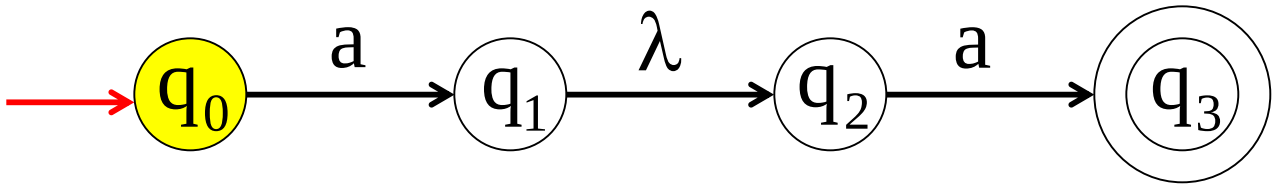
- Language accepted by the NFA: $L = \{aa\}$



Lambda Transitions

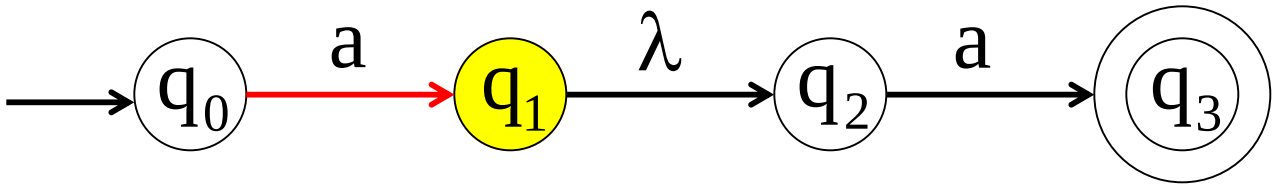


Lambda Transitions



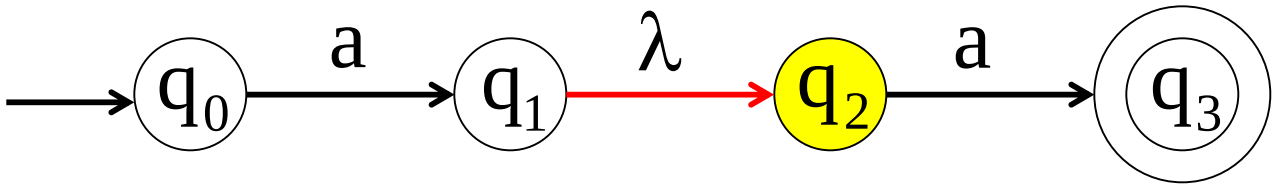
Lambda Transitions

Input String

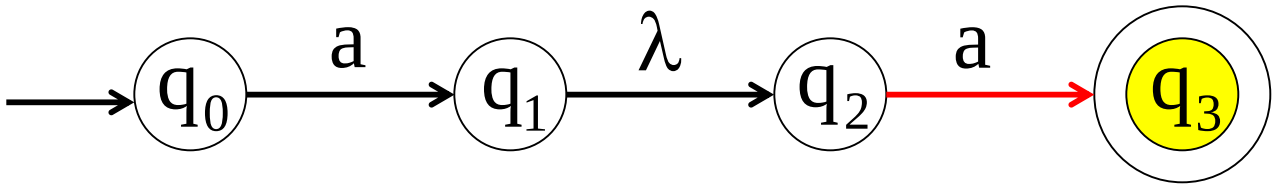


Lambda Transitions

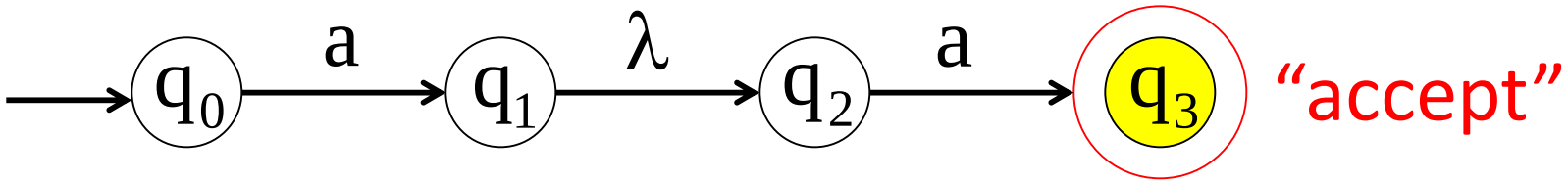
Read head does not move



Lambda Transitions

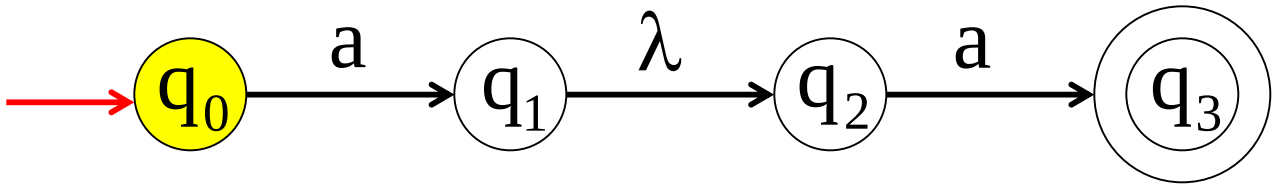


Lambda Transitions



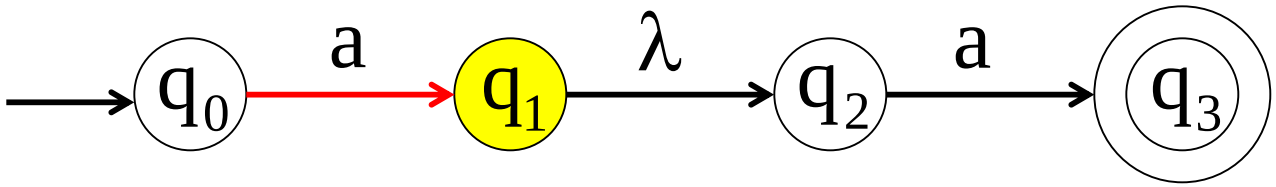
String aa is accepted

Rejection Example



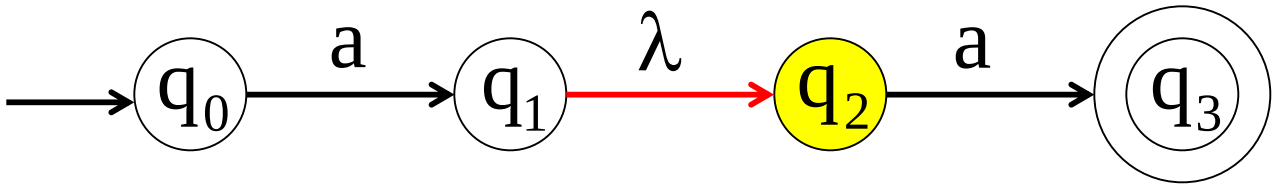
Rejection Example

Input String



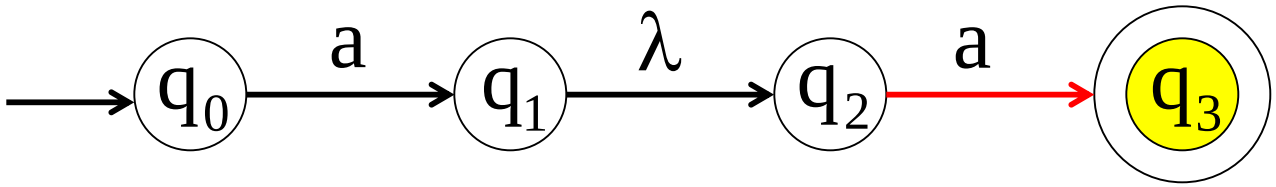
Rejection Example

Read head does not move

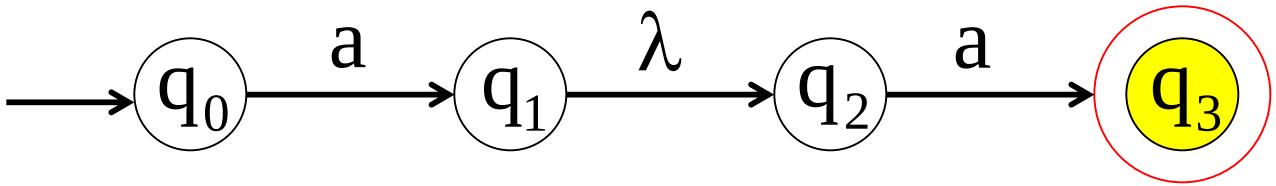


Rejection Example

Input String

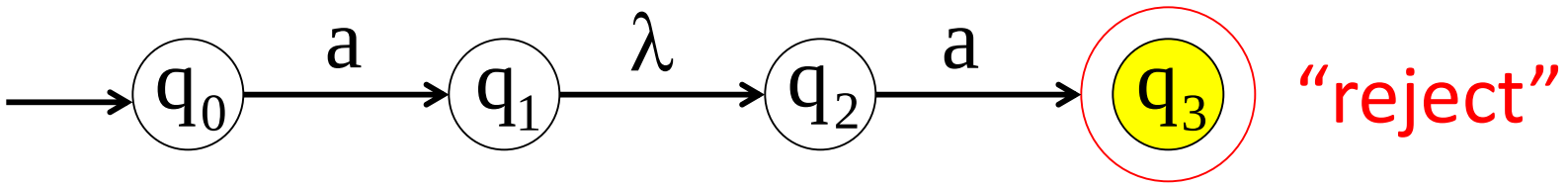


Rejection Example



No transition; NFA hangs

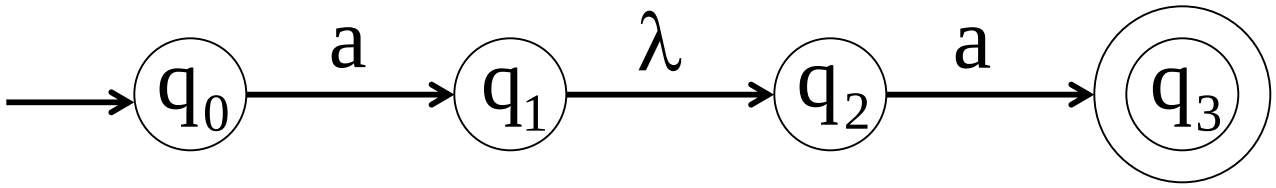
Rejection Example



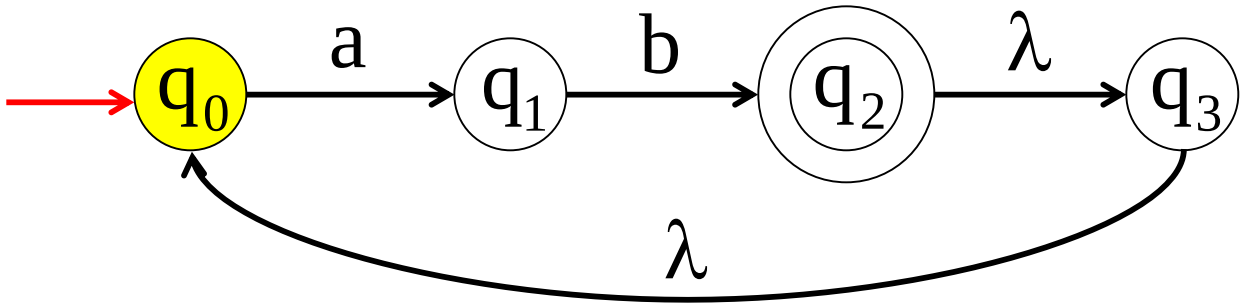
String aaa is rejected

Language

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

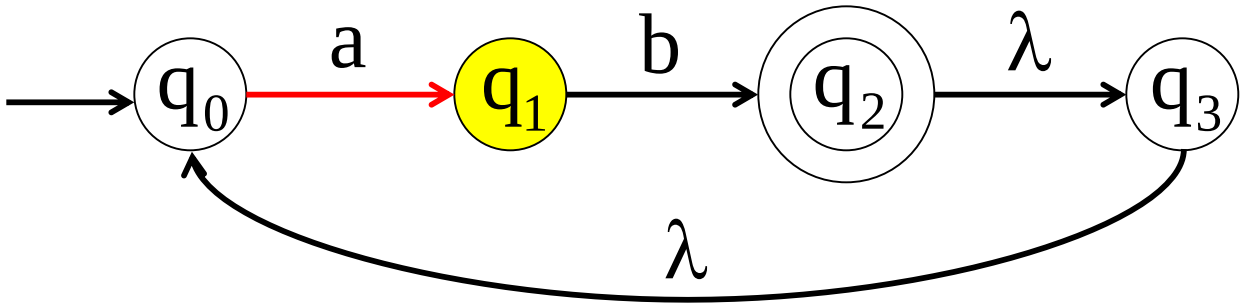


Another NFA Example



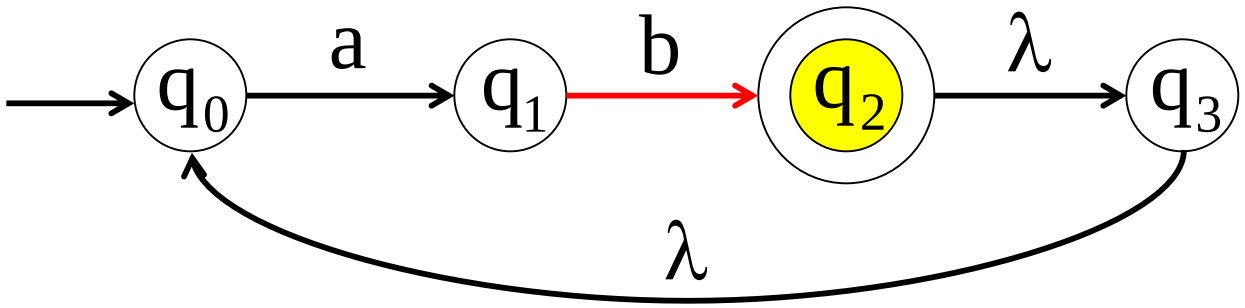
Another NFA Example

Input String

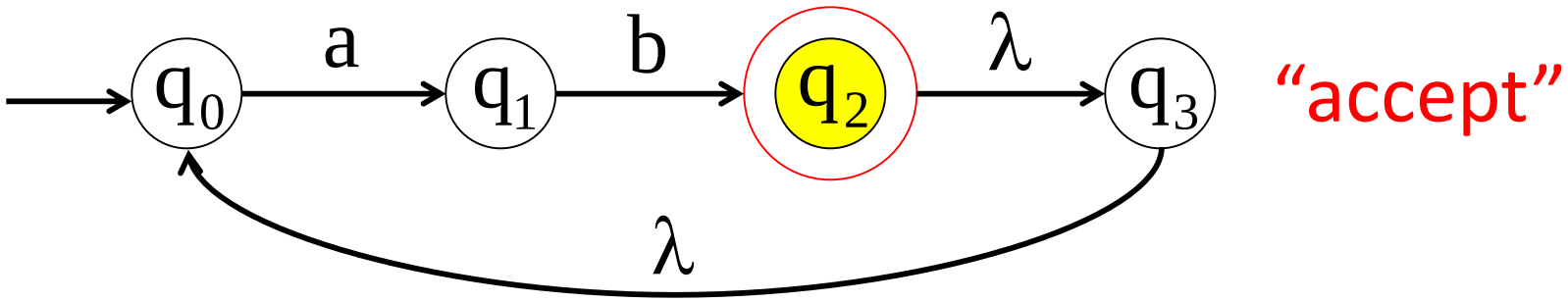


Another NFA Example

Input String

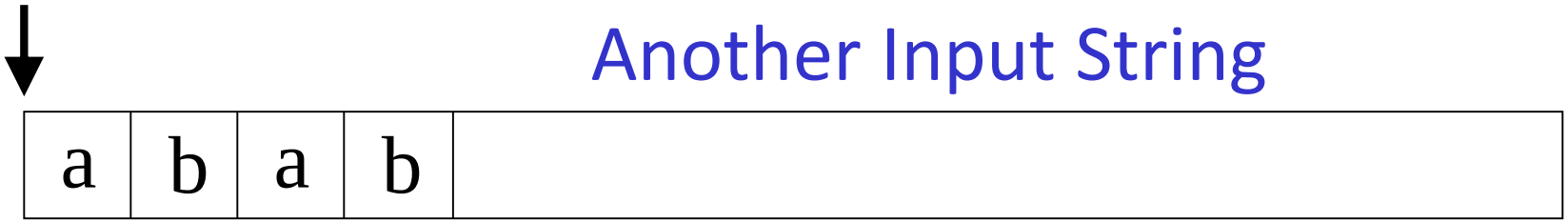


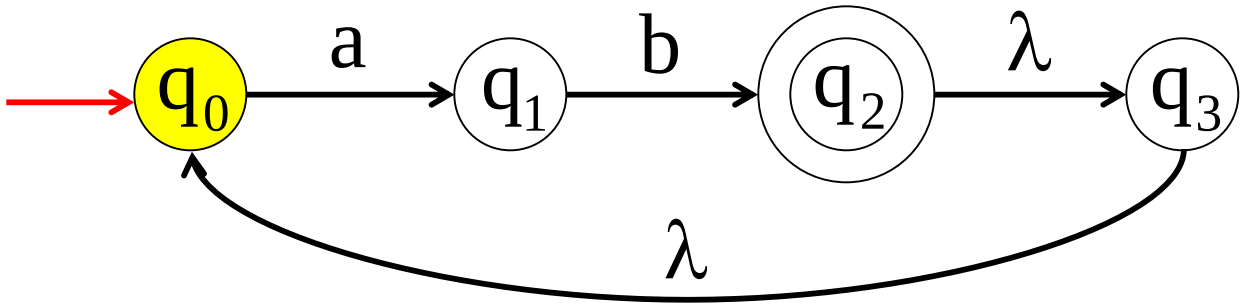
Another NFA Example



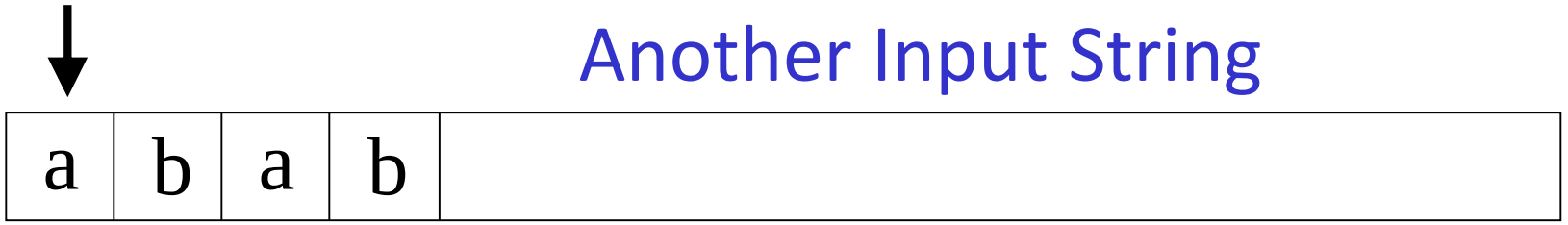
Another NFA Example

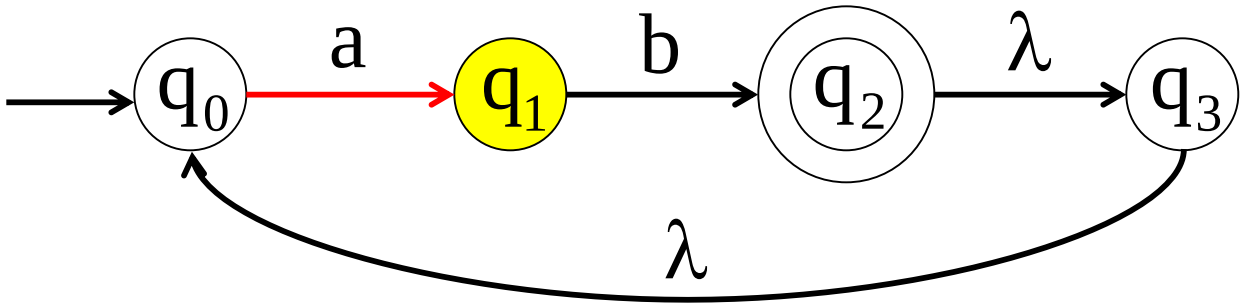
Another Input String





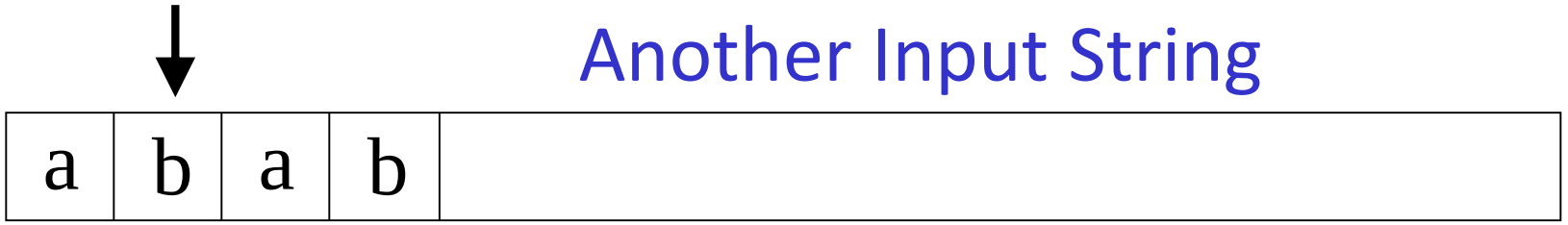
Another NFA Example

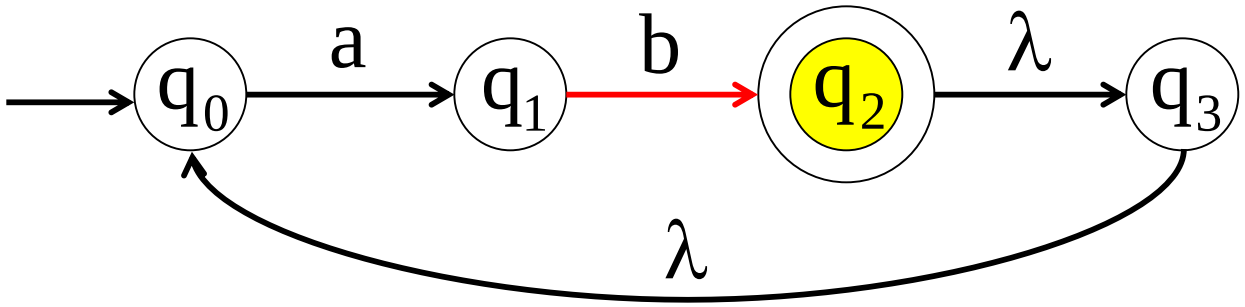




Another NFA Example

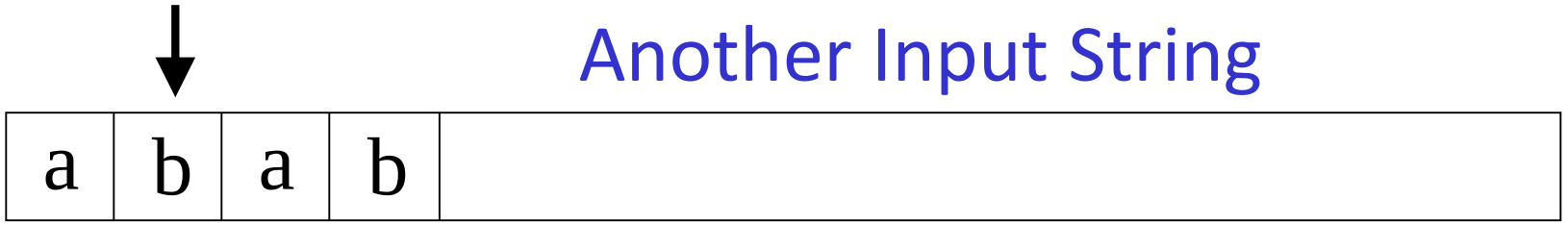
Another Input String

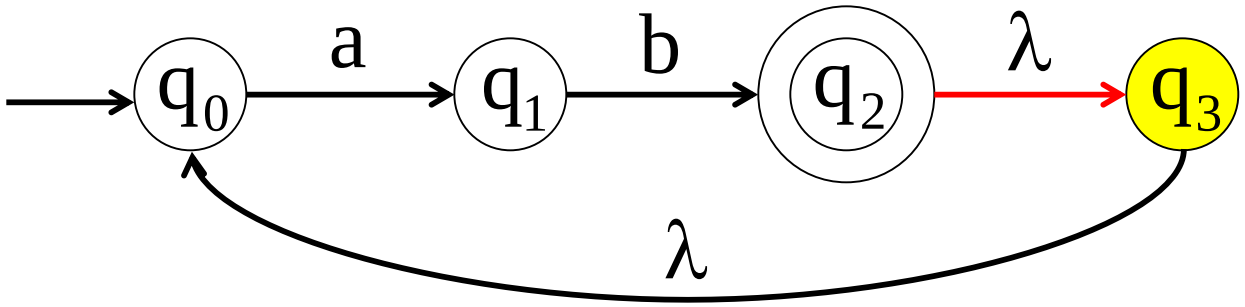




Another NFA Example

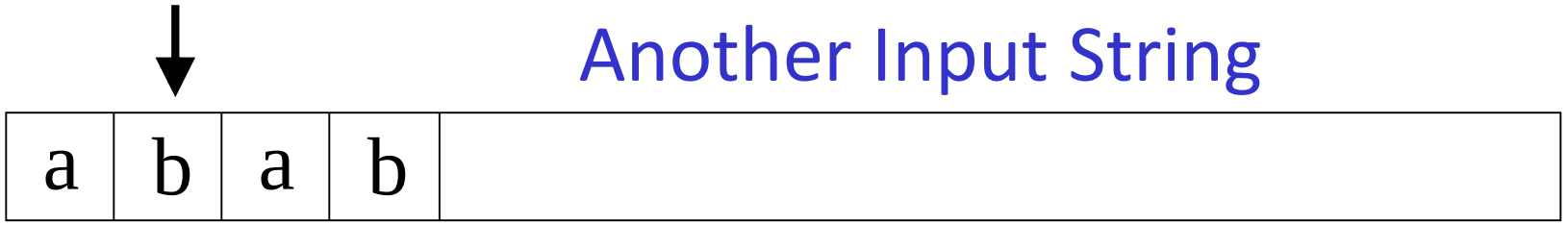
Another Input String

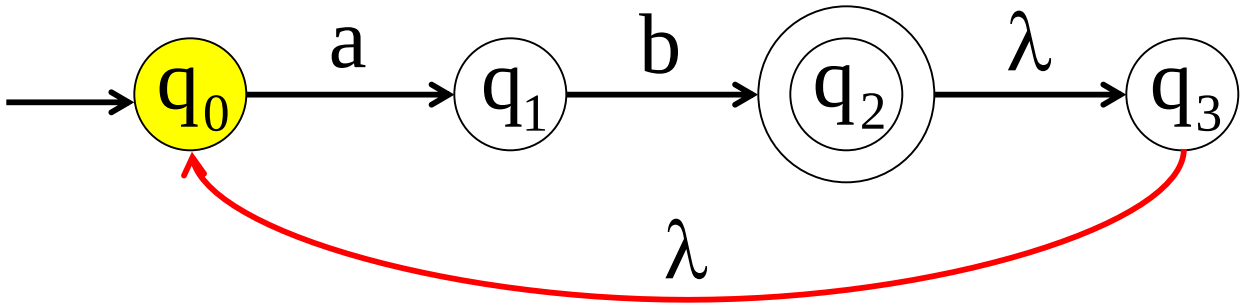




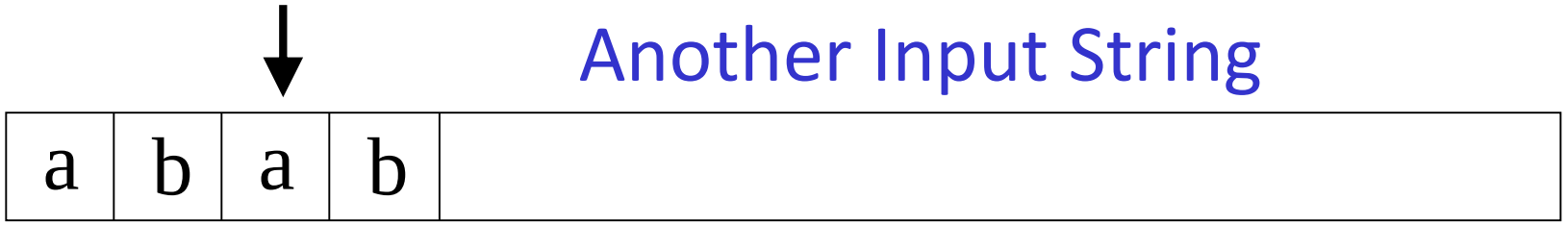
Another NFA Example

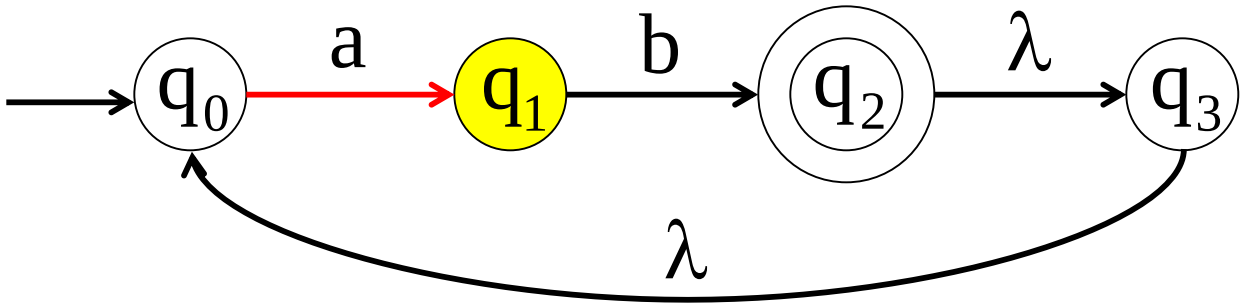
Another Input String





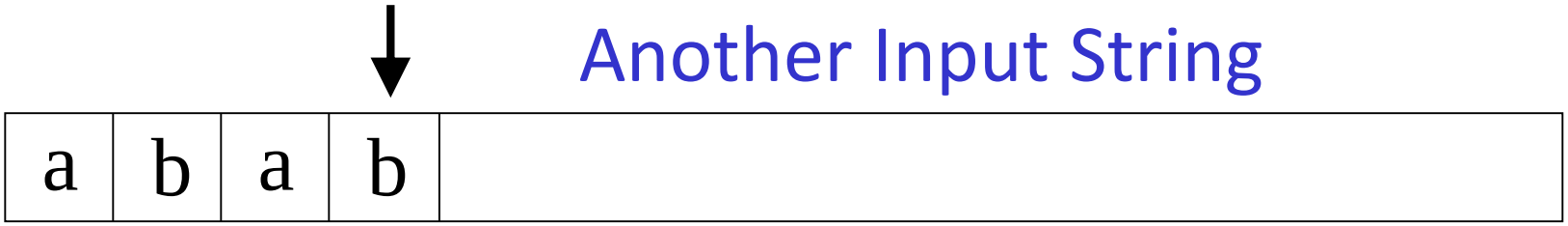
Another NFA Example

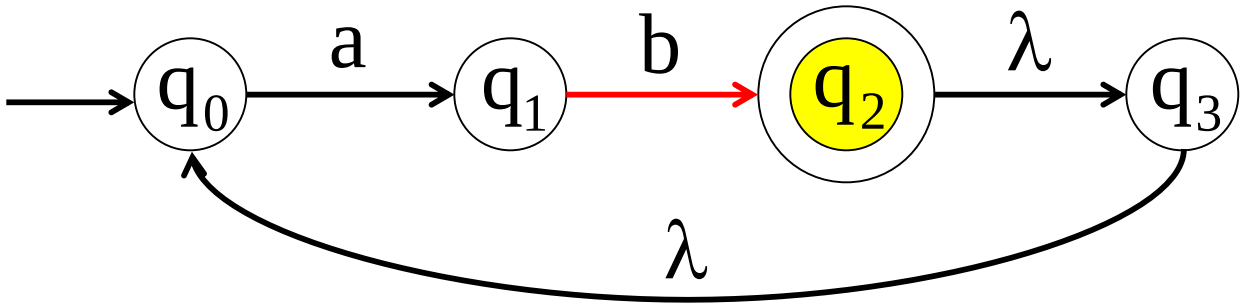




Another NFA Example

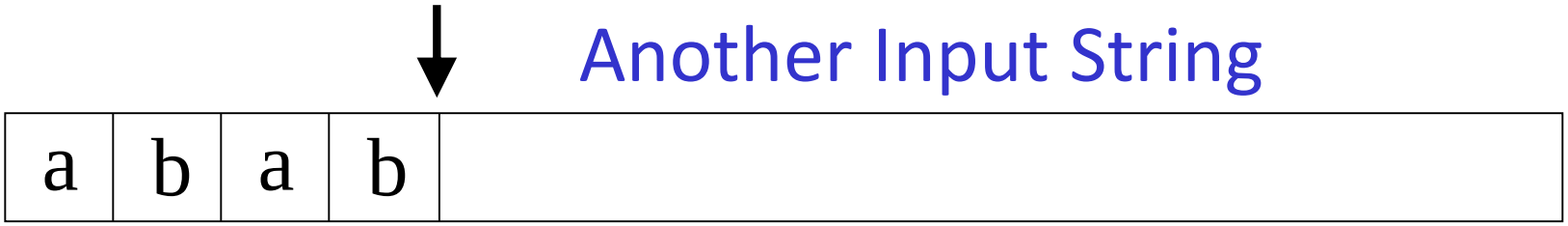
Another Input String

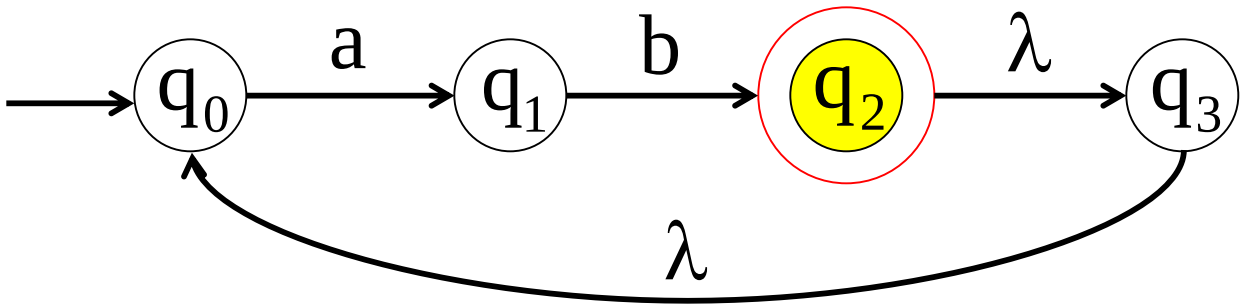




Another NFA Example

Another Input String

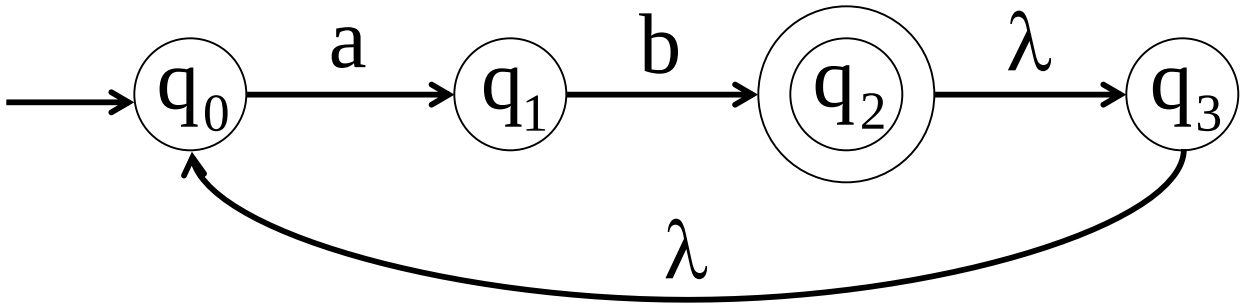




"accept"

Language Accepted

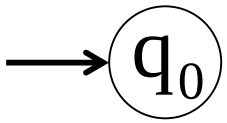
$$L = \{ab,\ abab,\ ababab,\ ...\} = \{ab\}^{+}$$



Remarks

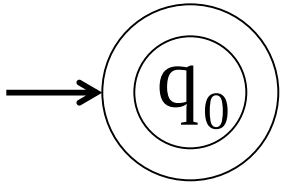
- The λ symbol never appears on the input tape
- Extreme automata:

M_1



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

M_2

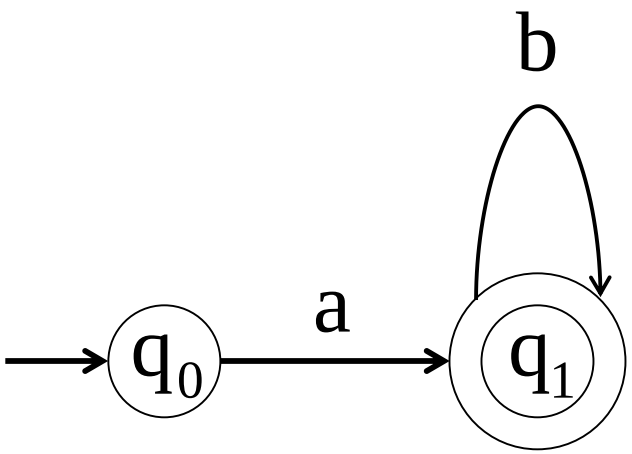


$L(M_2) = \{\lambda\}$

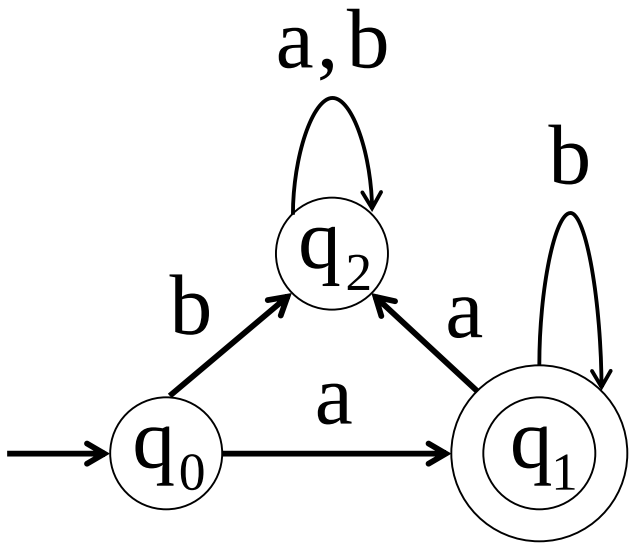
Remarks

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

NFA M_1



DFA M_2



Formal Definition of NFAs

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

Q: Set of states, i.e.

Σ: Input aplhabet, i.e.

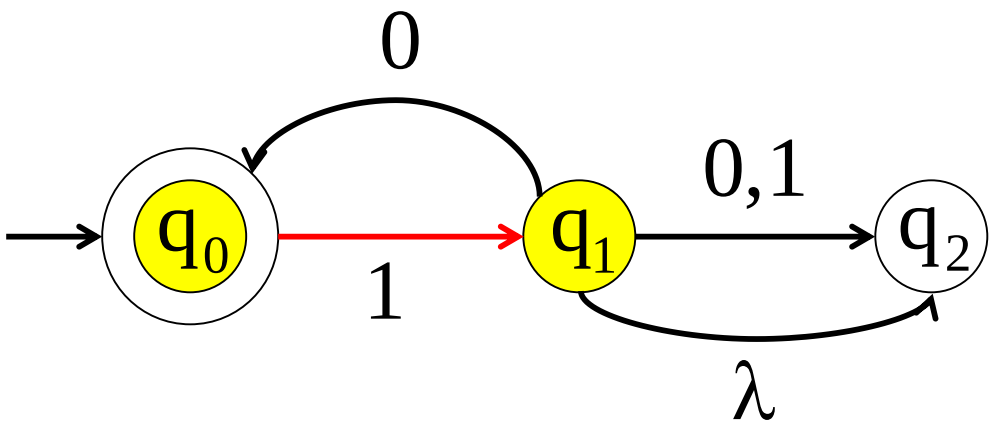
δ: Transition function

q_0: Initial state

F: Final states

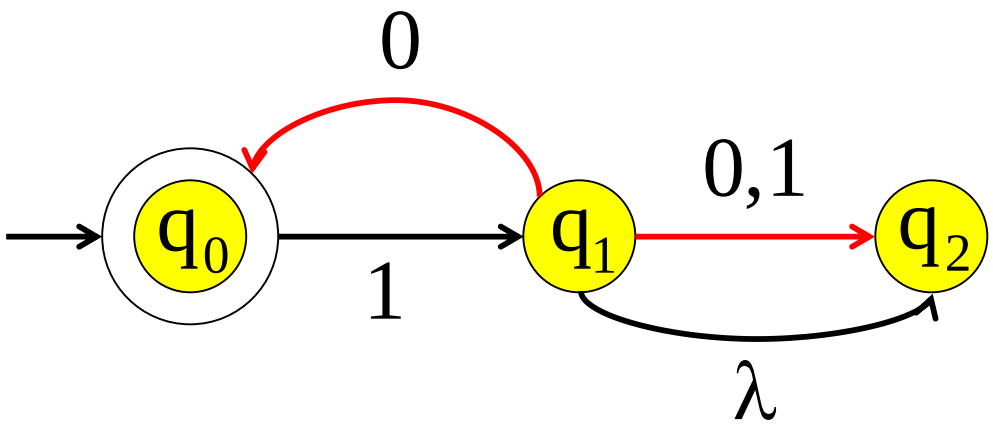
Transition Function δ

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



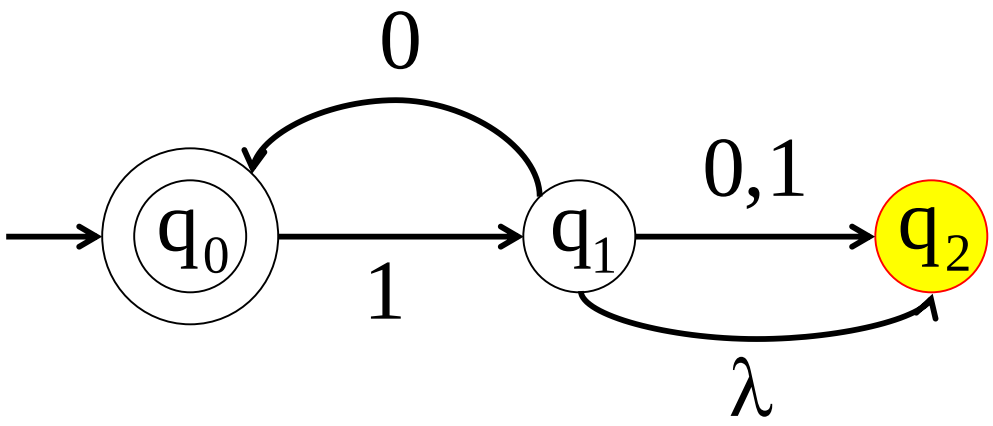
Transition Function δ

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



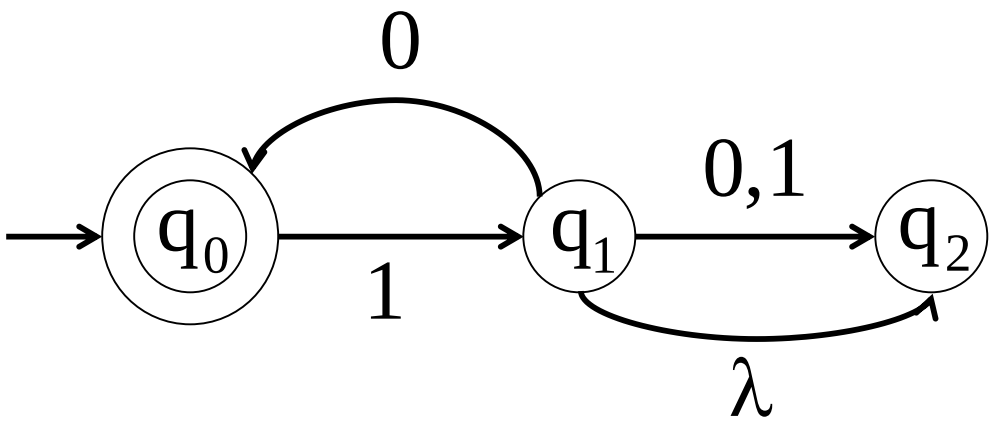
Transition Function δ

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



Language

Language: _____



Exercises

- For $\Sigma = \{0,1\}$, construct NFAs that accept the languages consisting of

1) all strings with **exactly one** 0.
2) all strings with **at least one** 0.
3) all strings with **no more than two** 0's.
4) all strings **starting with** 00.
5) all strings **ending with** 00.

Exercises

1. Design an NFA for the language

$$\{abab^n : n \geq 0\}$$

2. Design an NFA for the language

$$\{aba^n : n \geq 0\}$$

3. Design an NFA for the language

$$\{abab^n : n \geq 0\} \cup \{aba^n : n \geq 0\}$$

4. Design an NFA no more than 5 states for the language

$$\{abab^n : n \geq 0\} \cup \{aba^n : n \geq 0\}$$

Exercises

5. Construct an NFA that accepts the language

$$\{ab, abc\}^*$$

6. Construct an NFA with 3 states that accepts the language

$$\{ab, abc\}^*$$

7. Find an NFA that accepts the language

$$\{a^n : n \geq 1\} \cup \{b^m a^k : m, k \geq 0\}$$