Tianjin International Engineering Institute

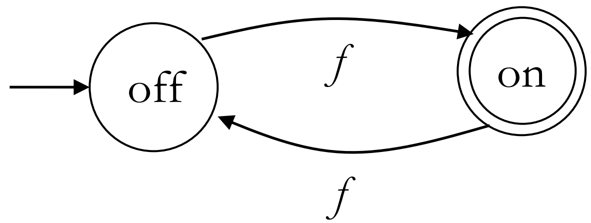
Formal Languages and Automata

Lesson 3: Finite Automata

Marc Gaetano

Edition 2018

Example of a finite automaton

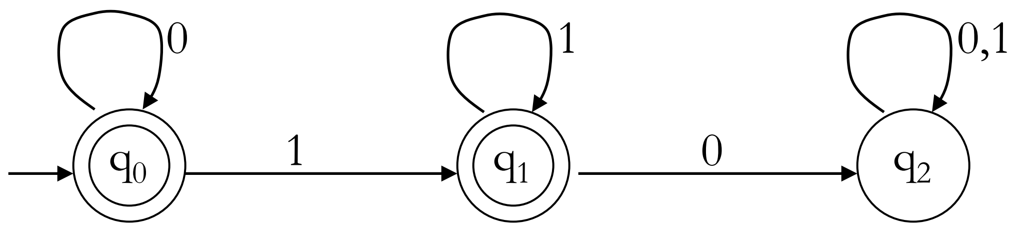


- There are states off and on, the automaton starts in off and tries to reach the "good state" on
- What sequences of fs lead to the good state?
- Answer: $\{f, fff, fffff, \ldots\} = \{f^{\,n} \colon n \text{ is odd}\}$
- This is an example of a deterministic finite automaton over alphabet $\{f\}$

Deterministic finite automata

- A deterministic finite automaton (DFA) is a 5-tuple $(\mathcal{Q}, \Sigma, \delta, q_0, F)$ where
  - $\mathcal{Q}$ is a finite set of states
  - Σ is an alphabet
  - $\delta: \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is a transition function
  - $q_0 \in \mathcal{Q}$ is the initial state
  - $F \subseteq \mathcal{Q}$ is a set of accepting states (or final states).
- In diagrams, the accepting states will be denoted by double loops

Example



alphabet $\Sigma = \{0, 1\}$
start state $Q = \{q_0, q_1, q_2\}$
initial state q_0
accepting states $F = \{q_0, q_1\}$

transition function δ:

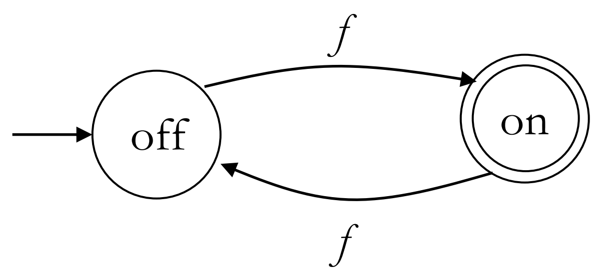
inputs

states	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

Language of a DFA

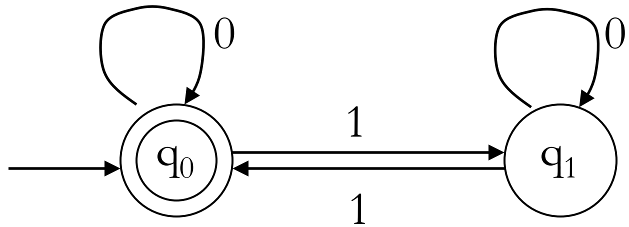
The language of a DFA $(Q, \Sigma, \delta, q_0, F)$ is the set of all strings over Σ that, starting from q_0 and following the transitions as the string is read left to right, will reach some accepting state

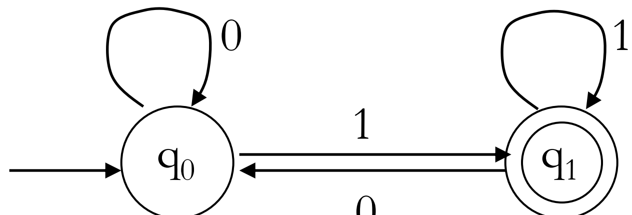
M:

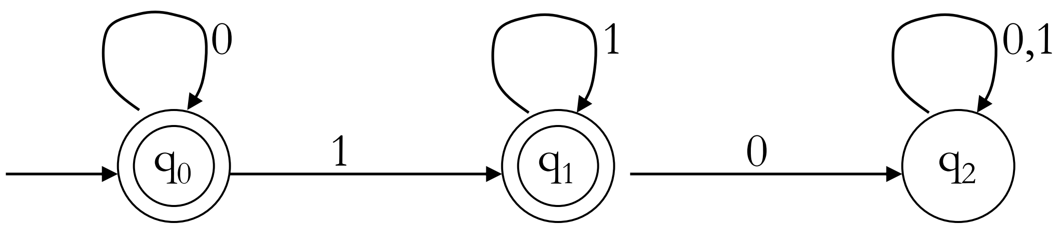


- Language of M is $\{f, fff, fffff, \ldots\} = \{f^{\,n}\colon n \text{ is odd}\}$

Examples







What are the languages of these DFAs?

Examples

- Construct a DFA that accepts the language

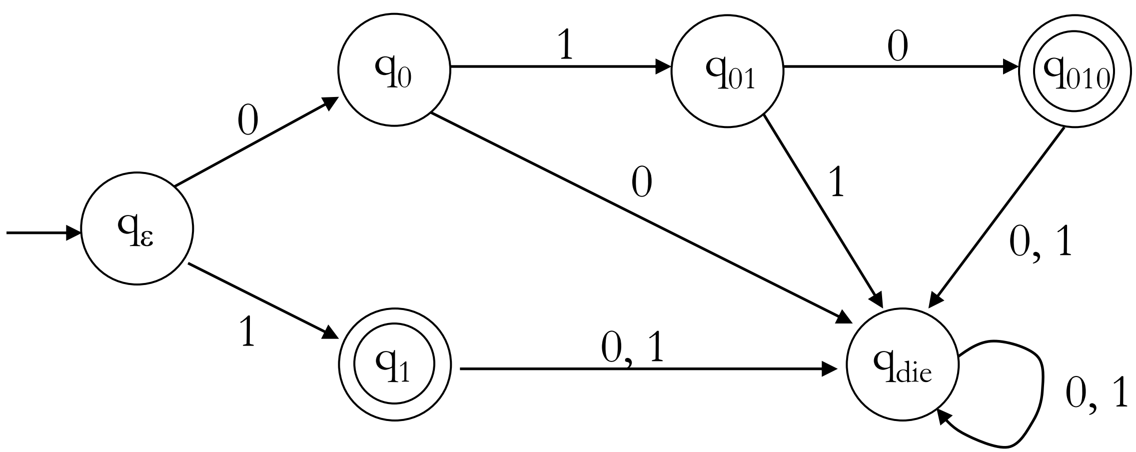
$$L = \{010, 1\} \qquad (\Sigma = \{0, 1\})$$

Examples

- Construct a DFA that accepts the language

$$L = \{010, 1\} \qquad\qquad (\,\Sigma = \{0, 1\}\,)$$

- Answer



Examples

- Construct a DFA over alphabet $\{0, 1\}$ that accepts all strings that end in 101

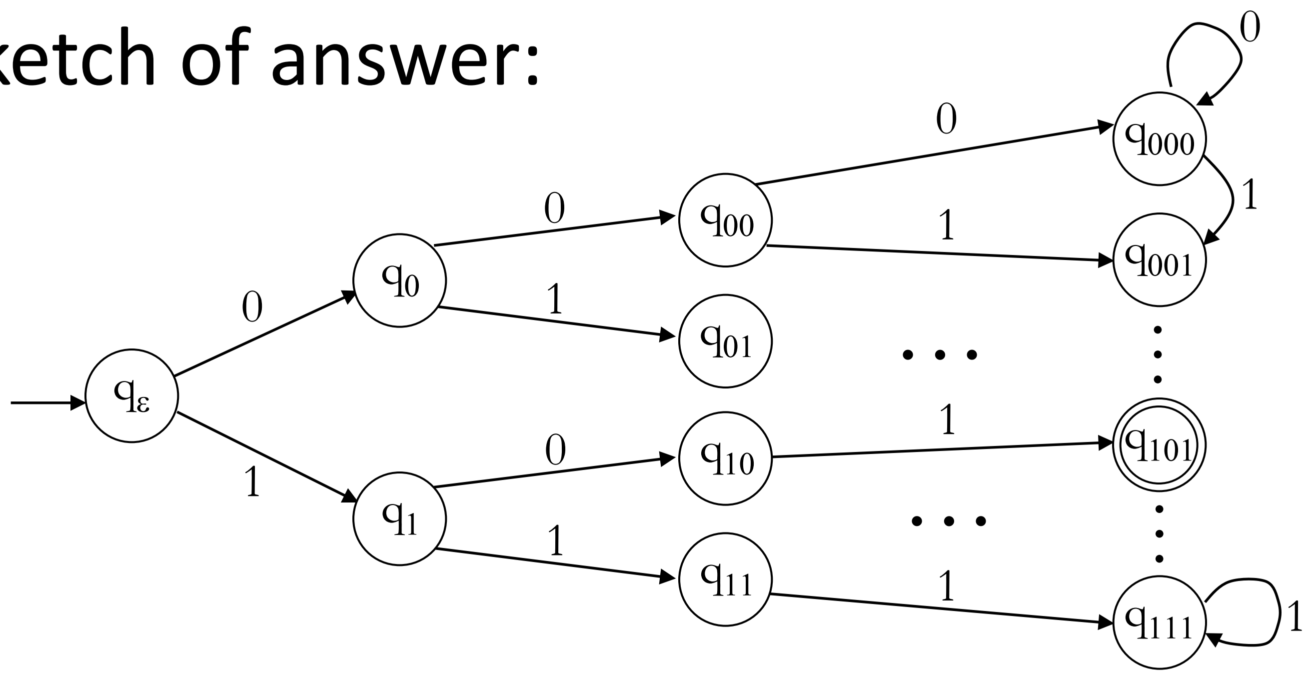
Examples

- Construct a DFA over alphabet $\{0, 1\}$ that accepts all strings that end in 101

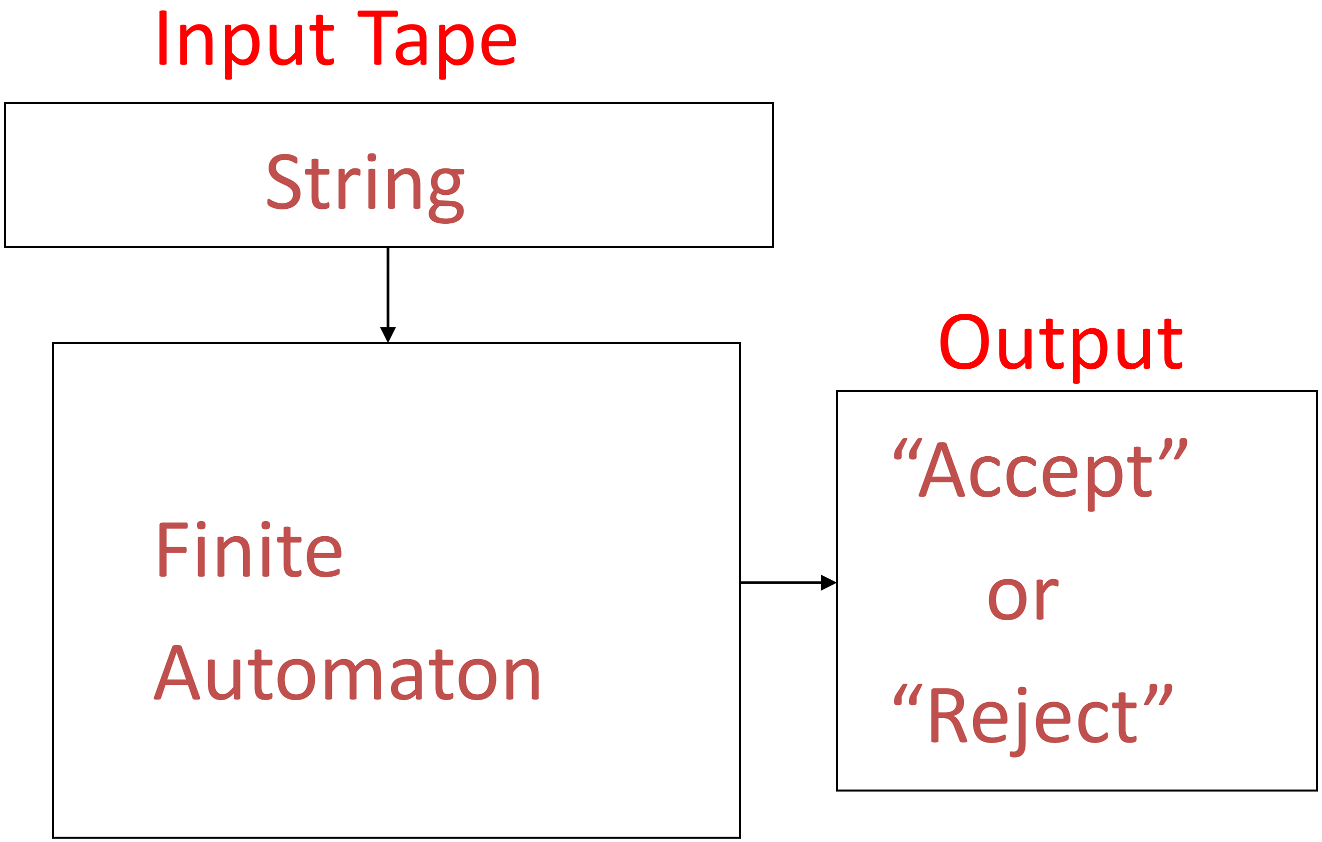
- Hint: The DFA must "remember" the last 3 bits of the string it is reading

Examples

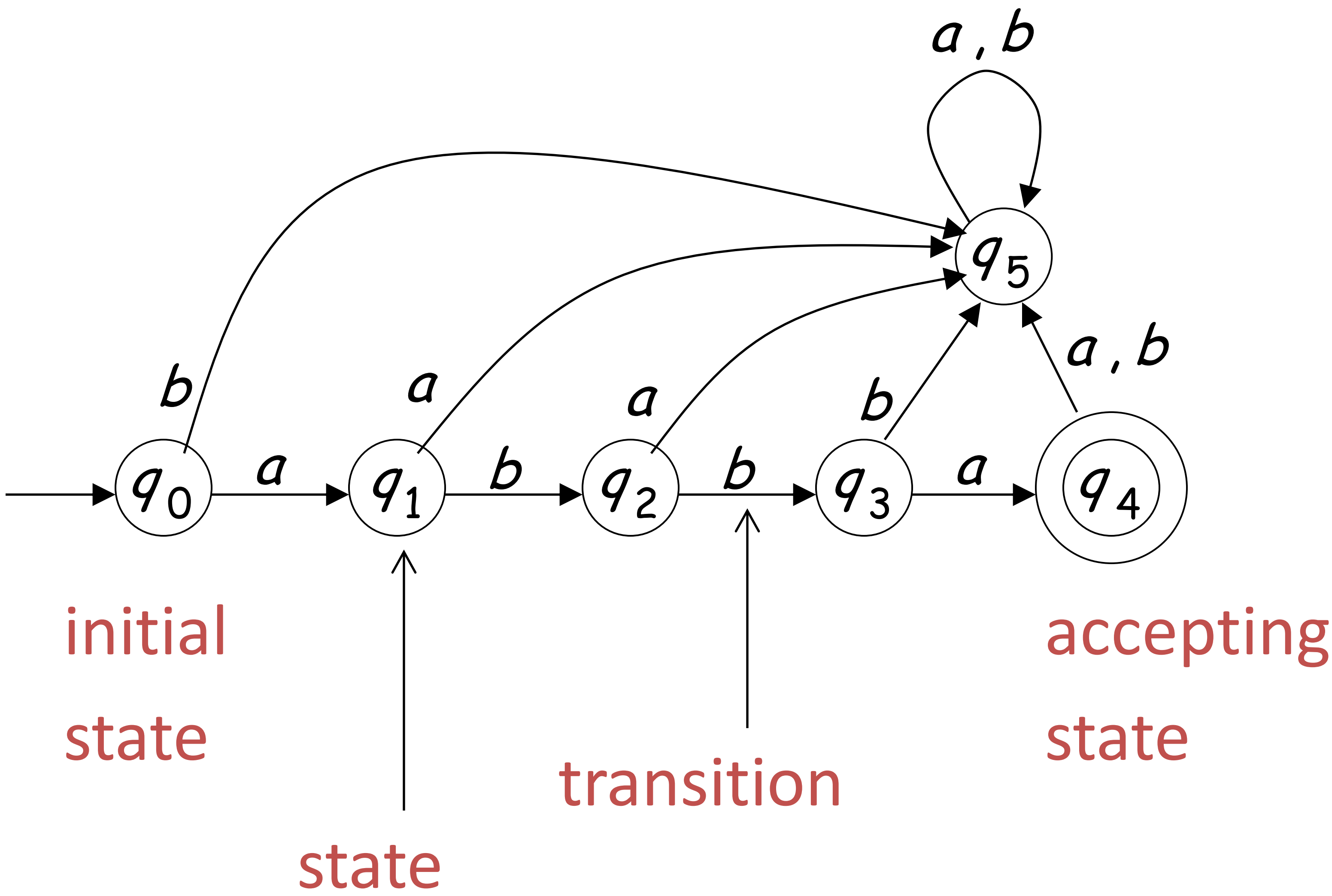
- Construct a DFA over alphabet $\{0, 1\}$ that accepts all strings that end in 101
- Sketch of answer:



DFA processing

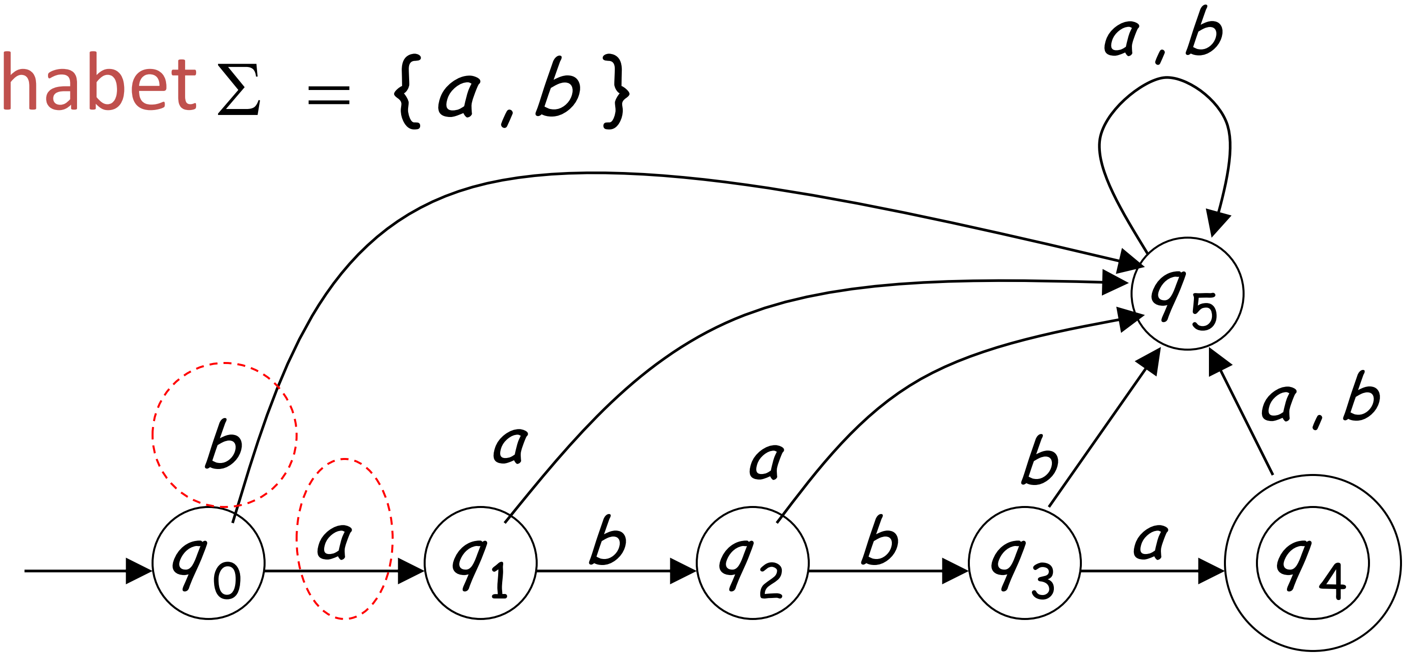


Transition Graph



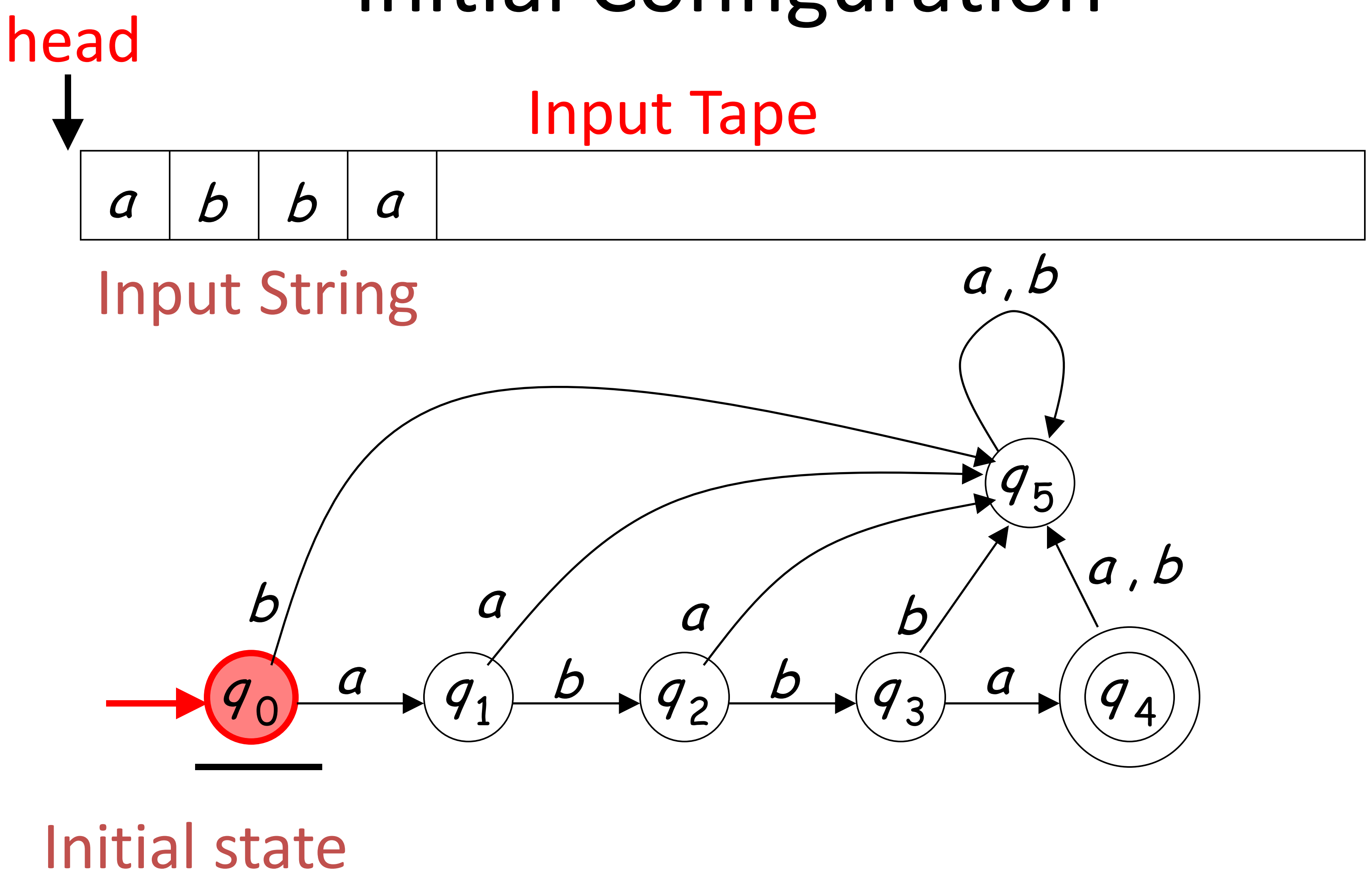
Transition Graph

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

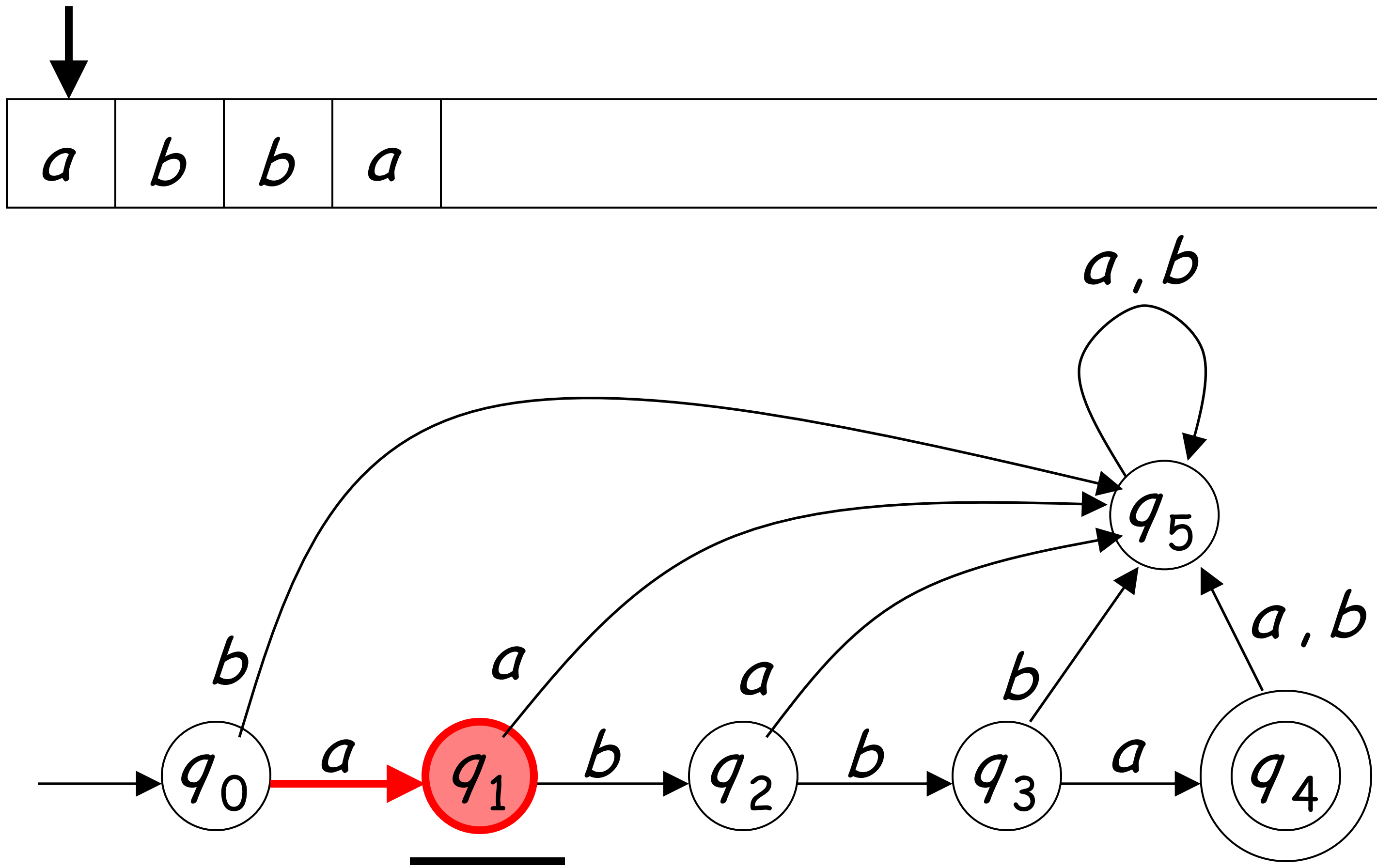


For <u>every</u> state, there is a transition for <u>every</u> symbol in the alphabet

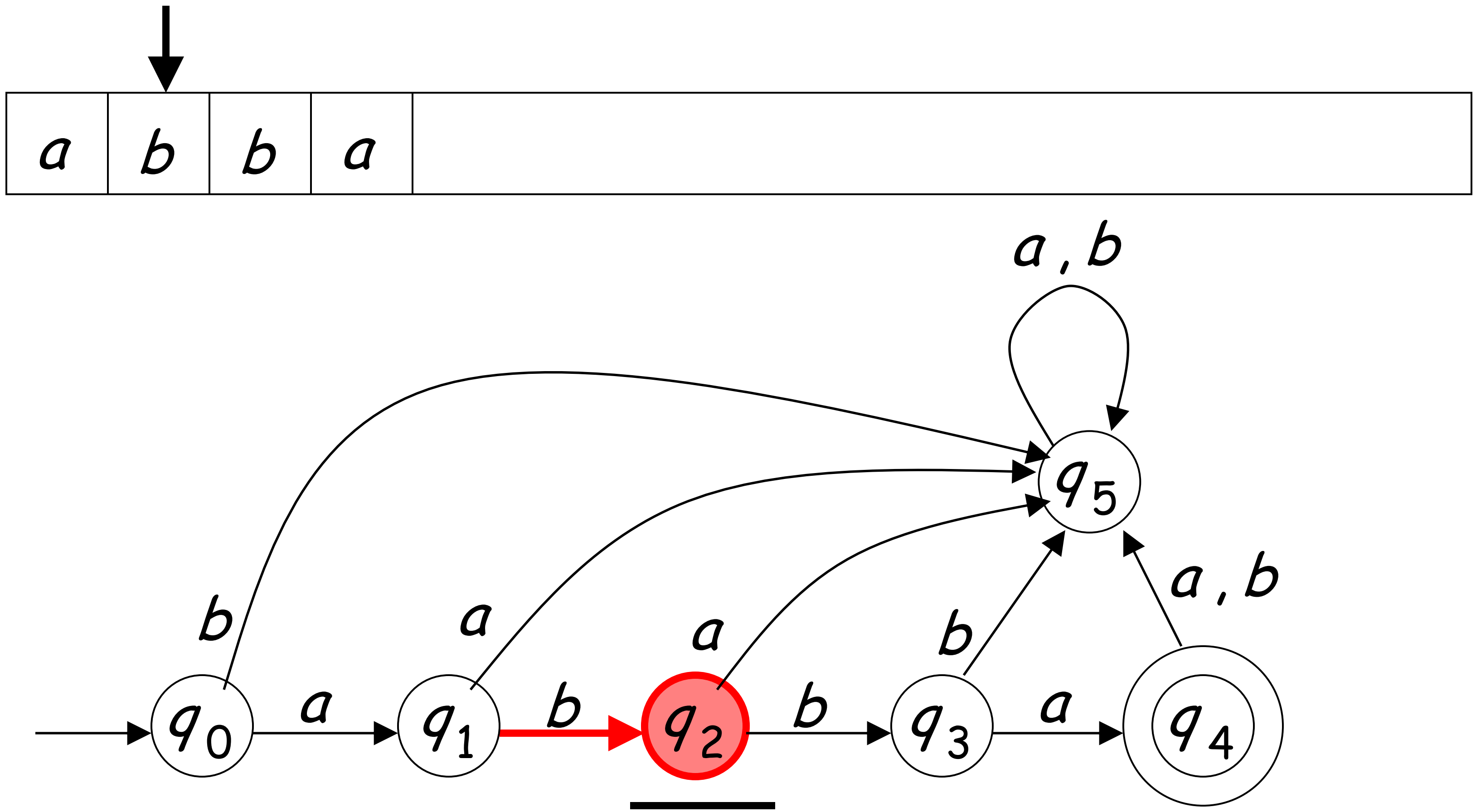
Initial Configuration



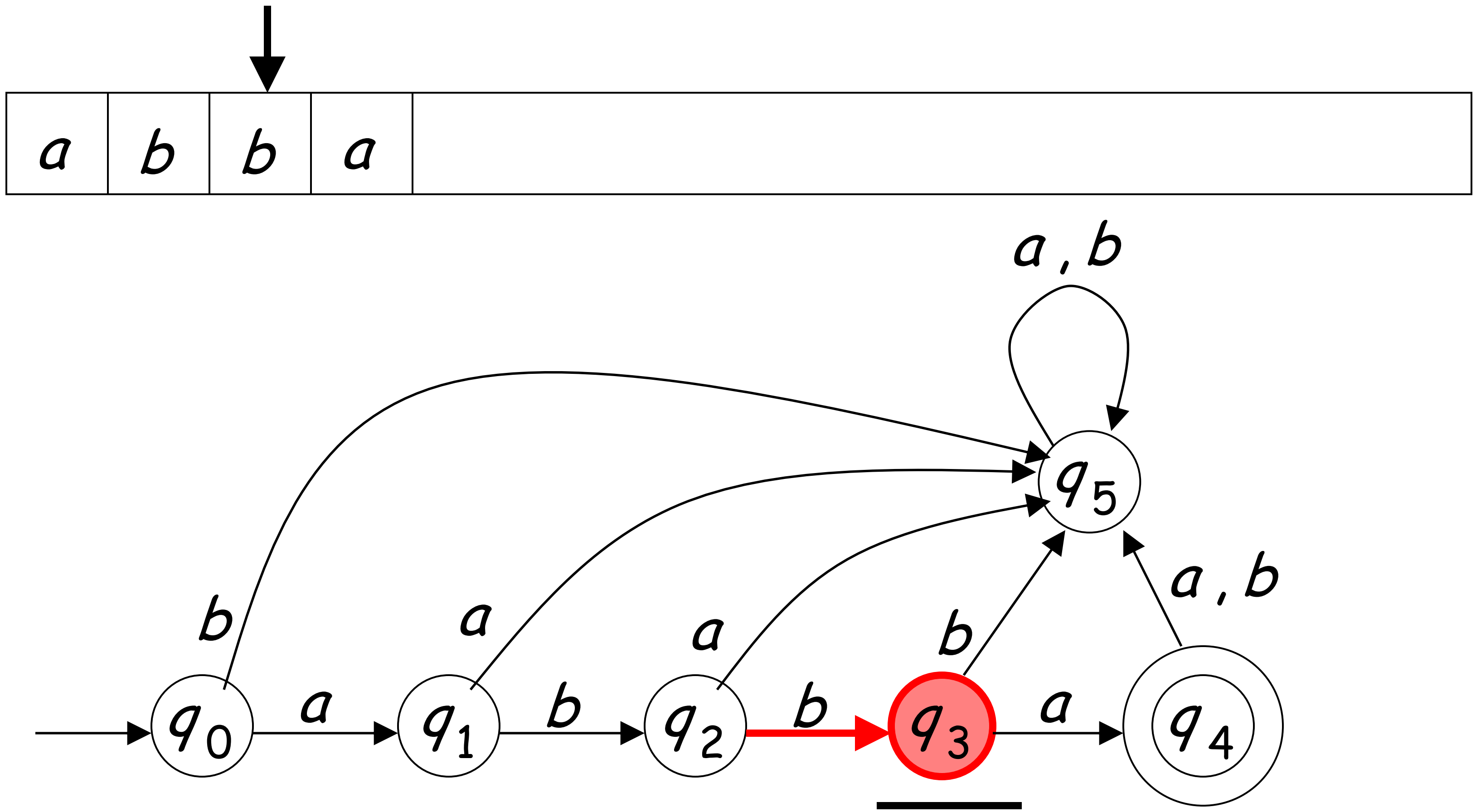
Scanning the Input



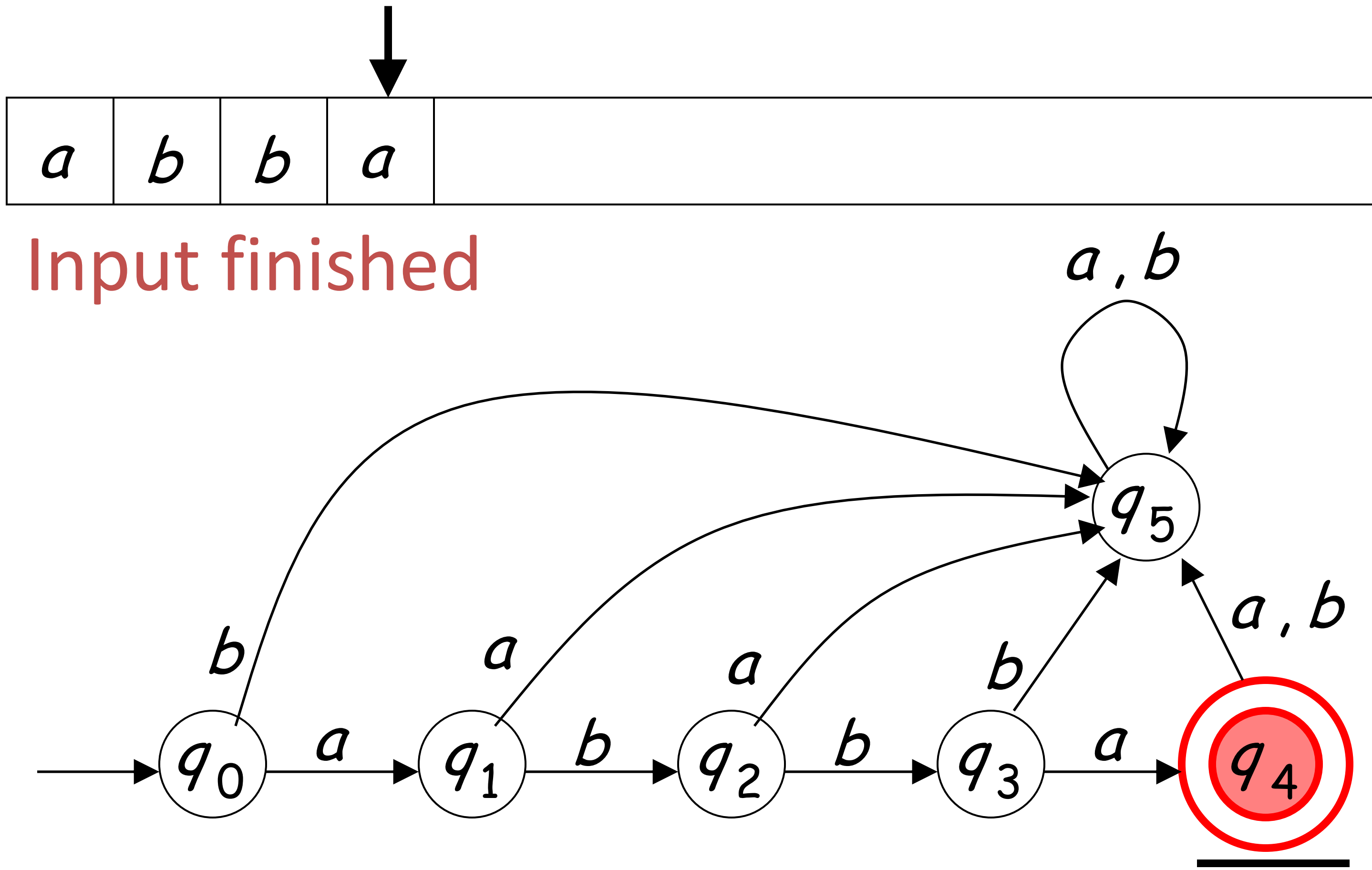
Scanning the Input



Scanning the Input



Scanning the Input



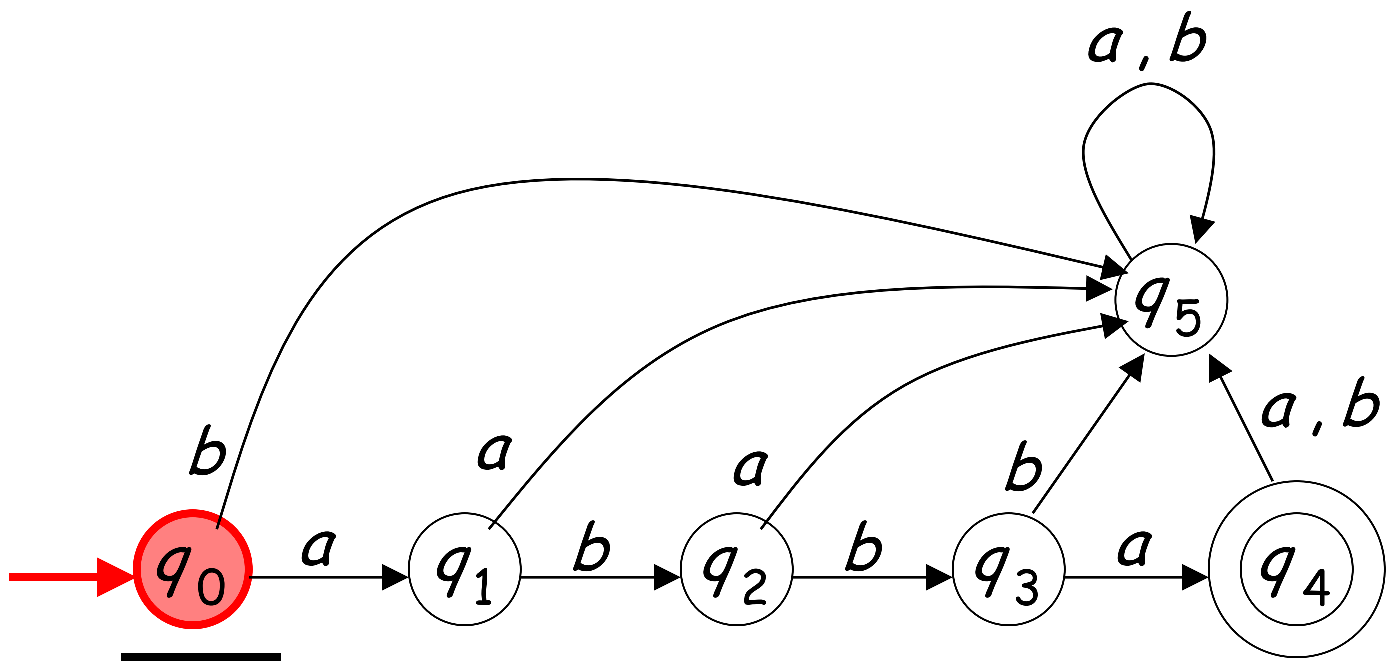
Input finished

accept

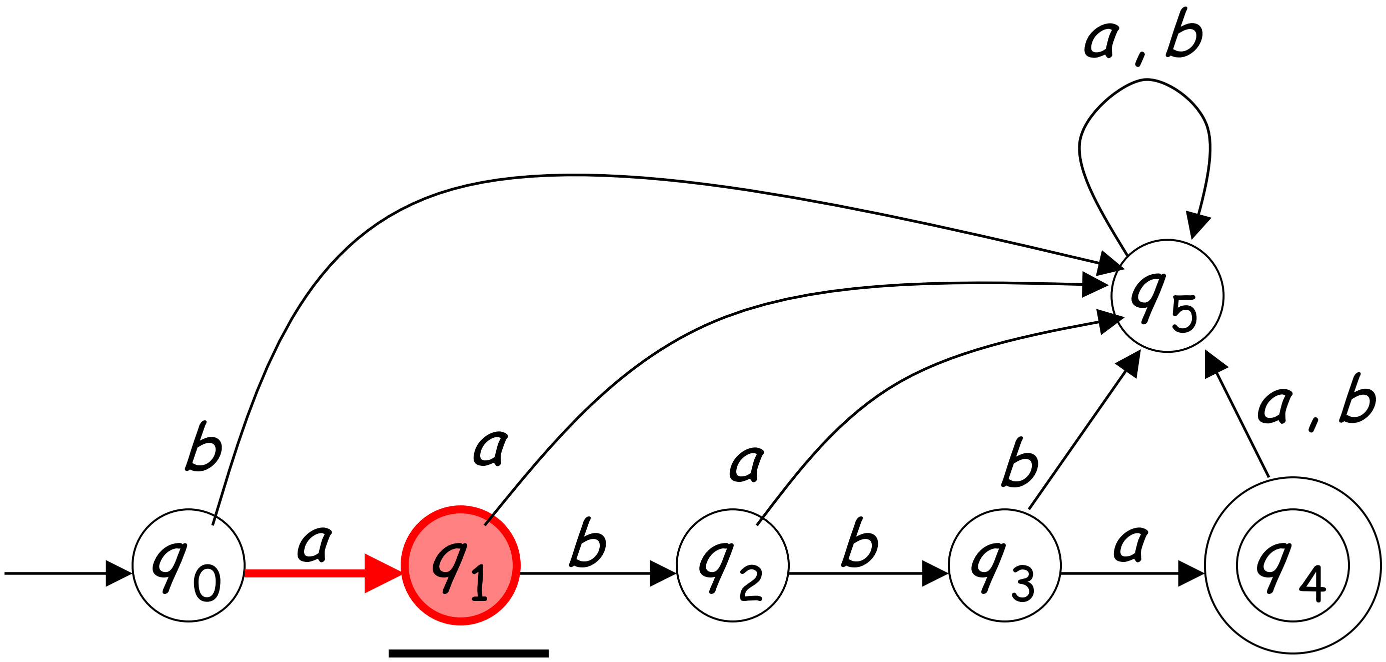
Last state determines the outcome

A rejection case

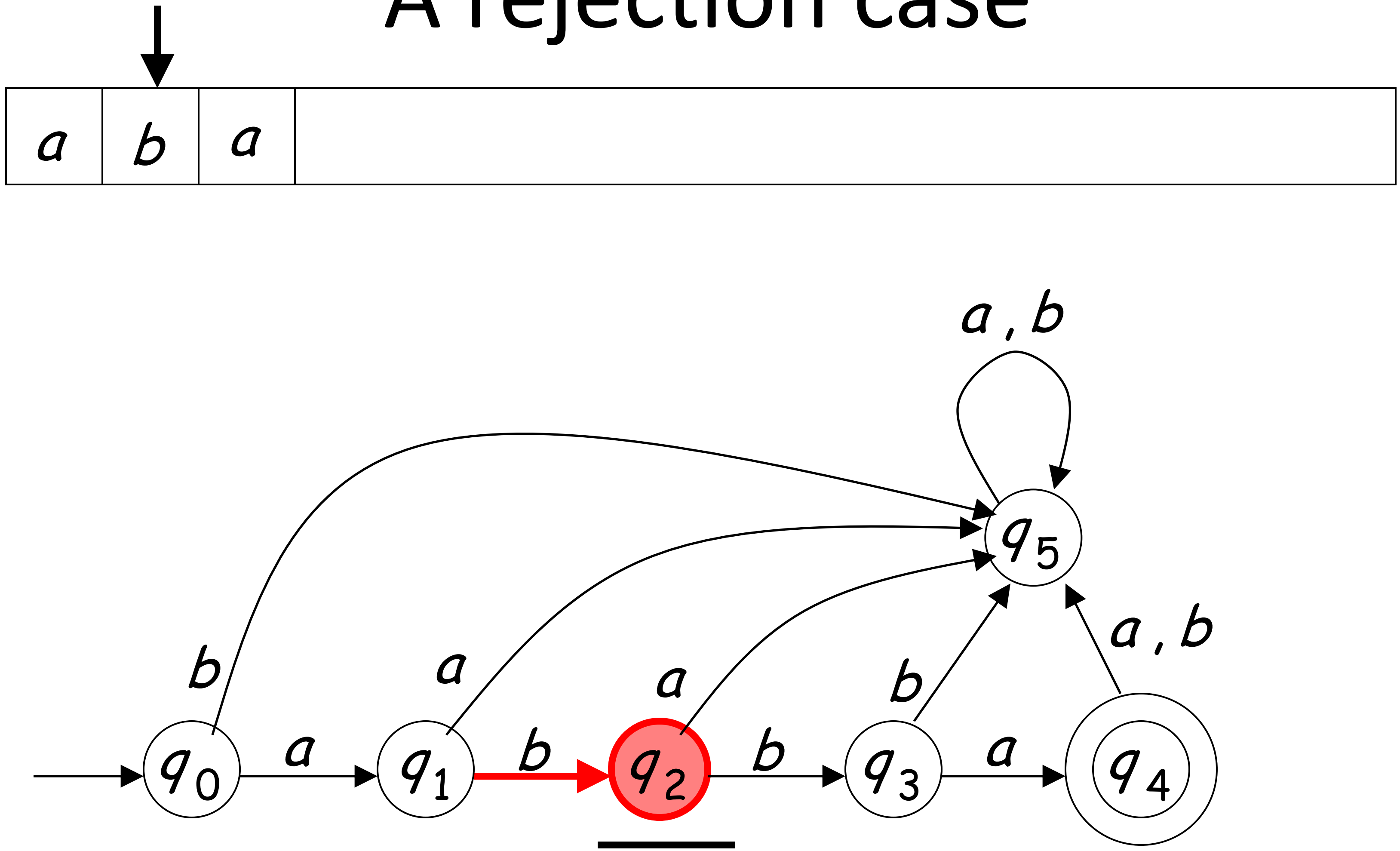
Input String



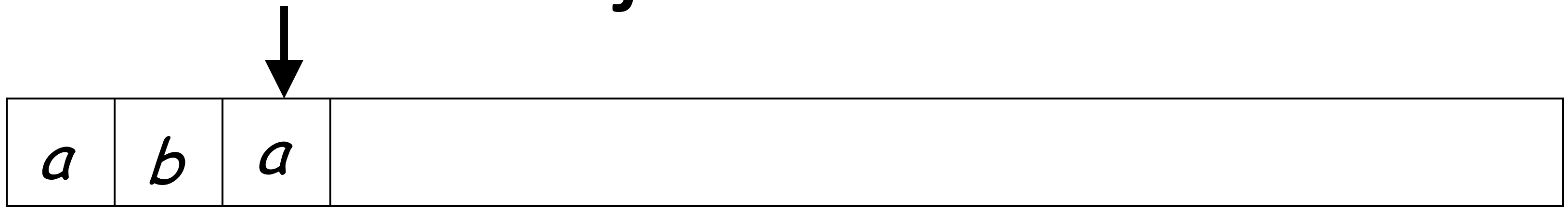
A rejection case



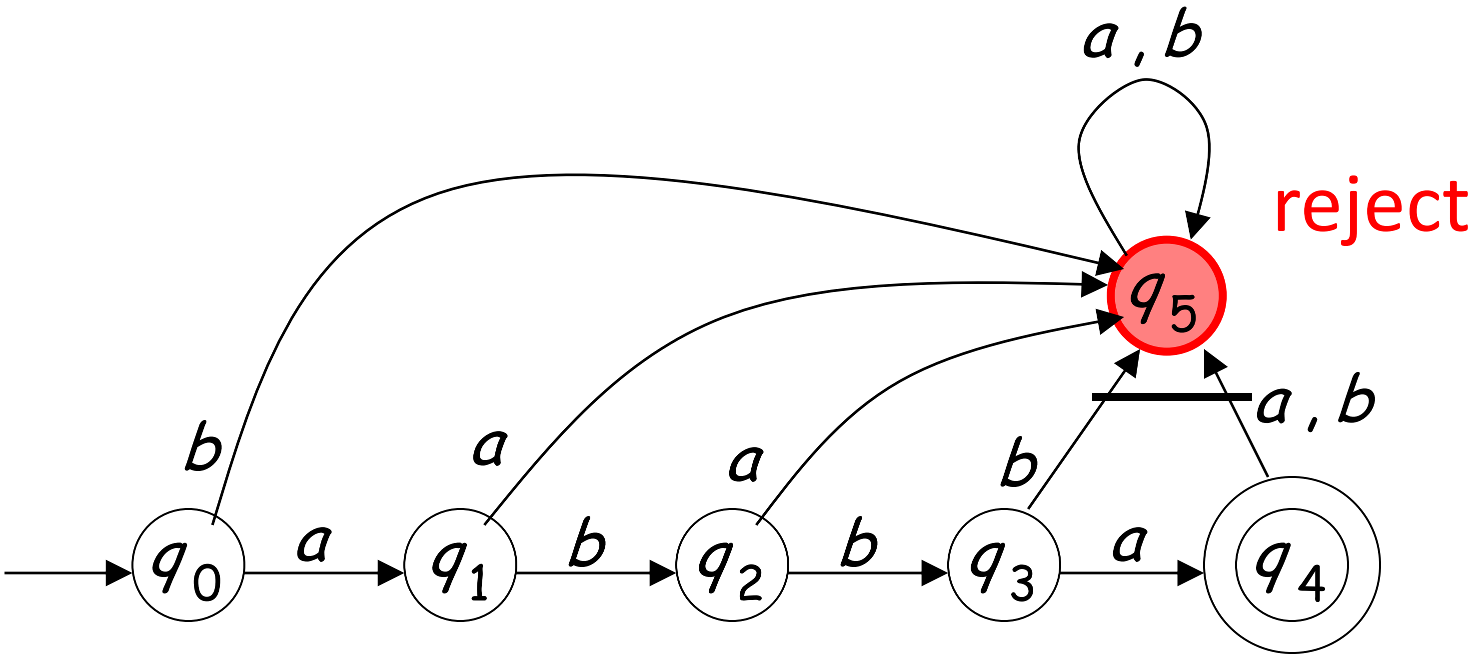
A rejection case



A rejection case



Input finished



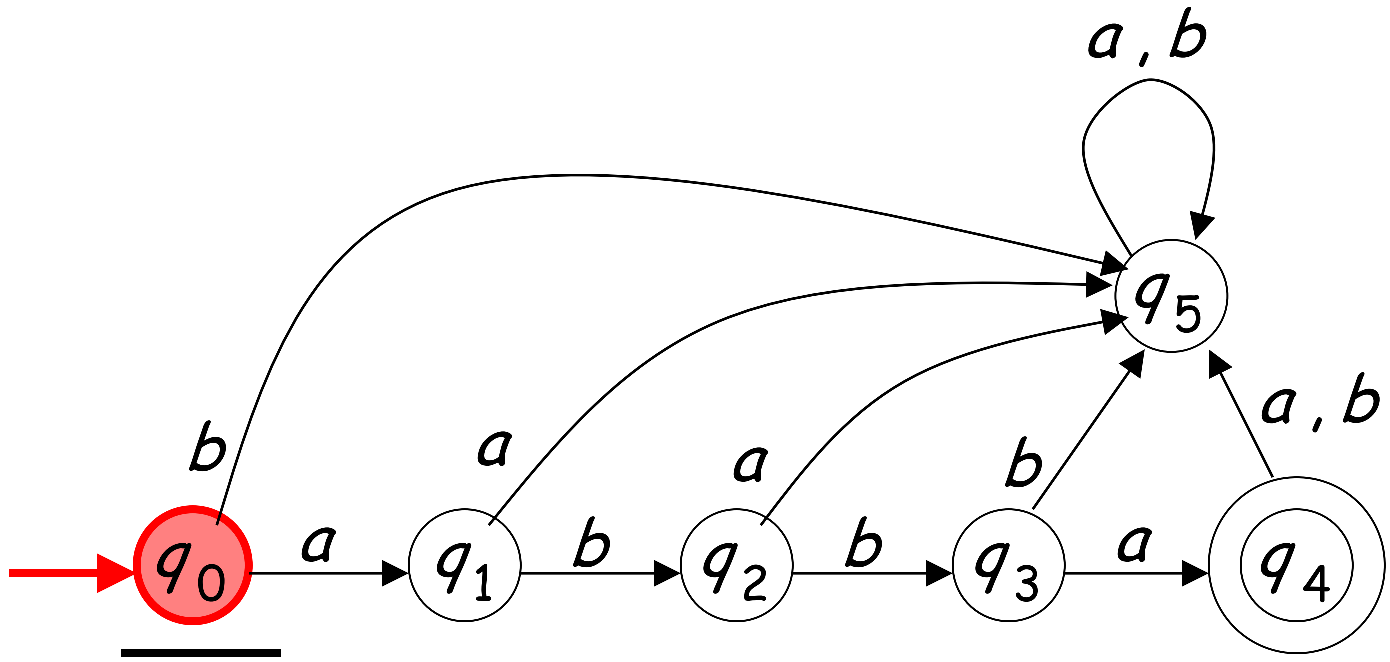
Last state determines the outcome

Another rejection case

(λ)

Input Finished (no symbol read)

Tape is empty



reject

Acceptation/Rejection

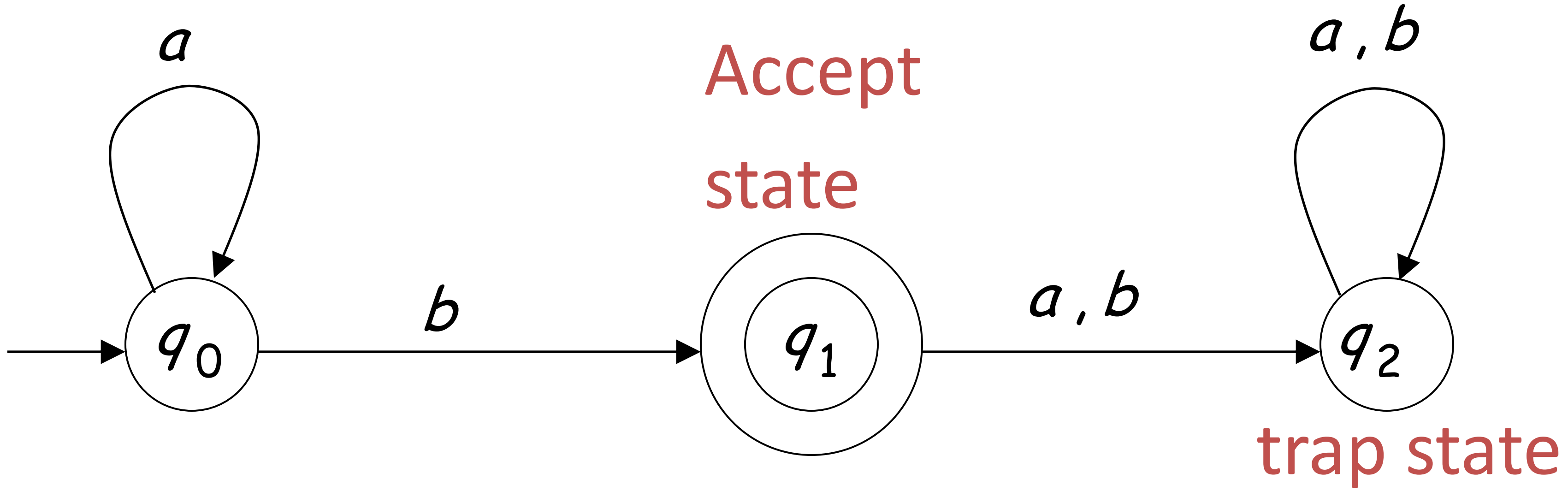
To accept a string:

all the input string is scanned
and the last state is accepting

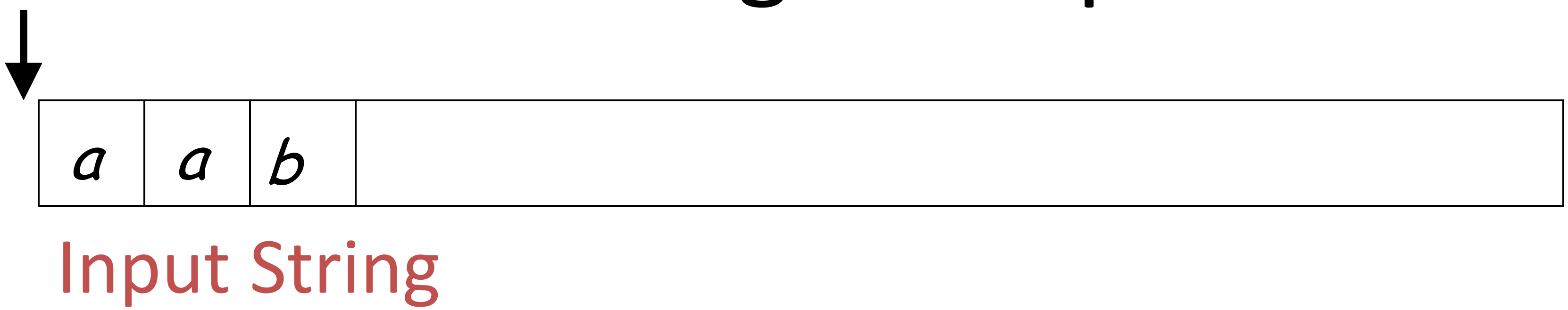
To reject a string:

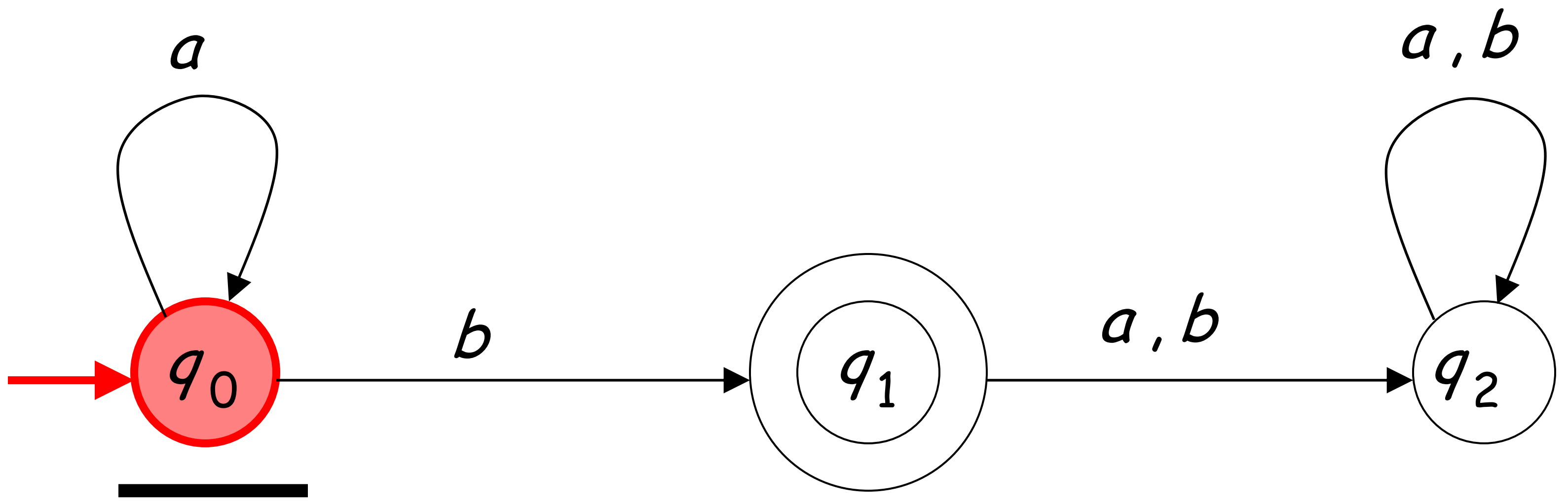
all the input string is scanned
and the last state is non-accepting

Another Example

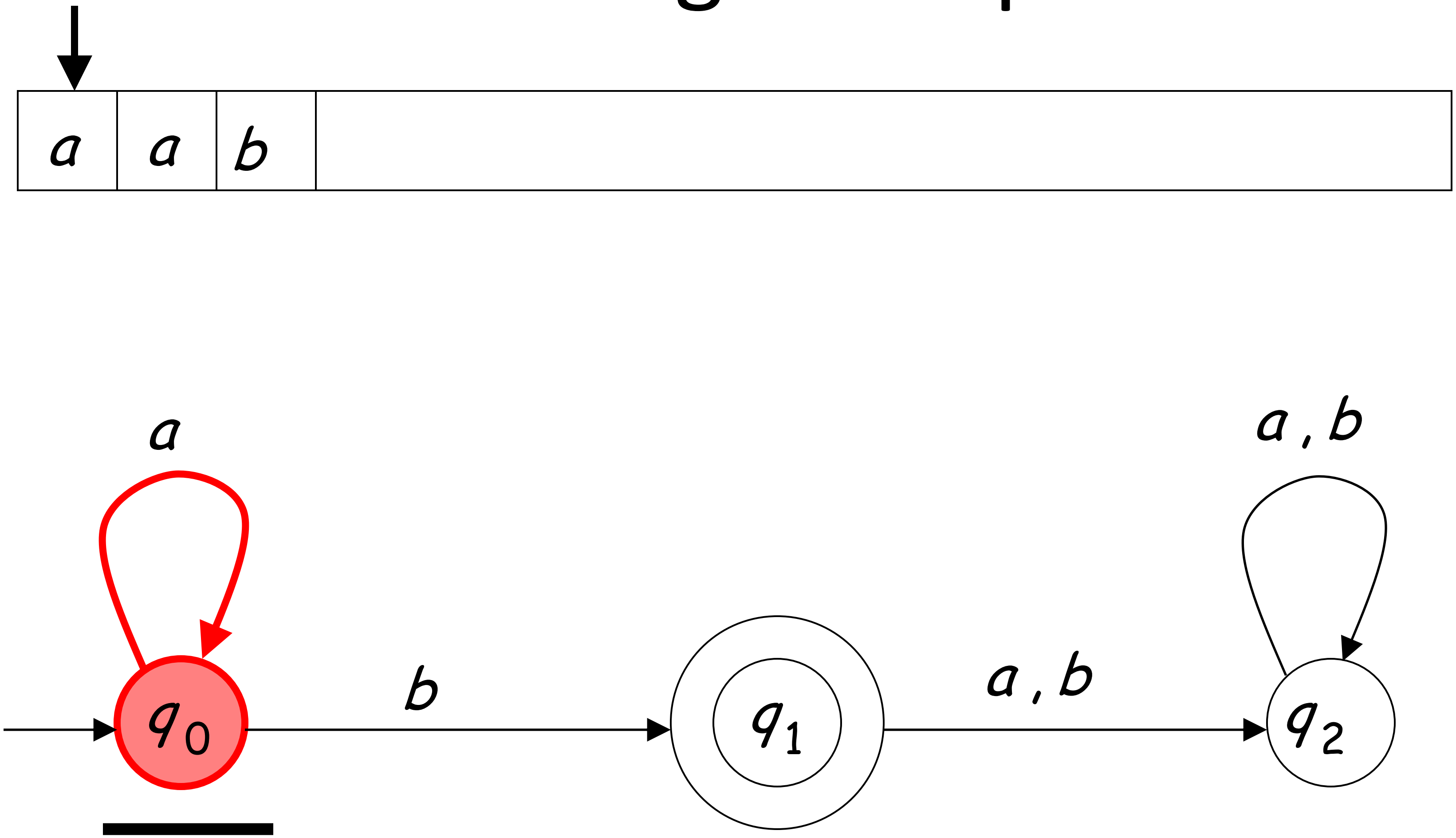


Scanning the Input

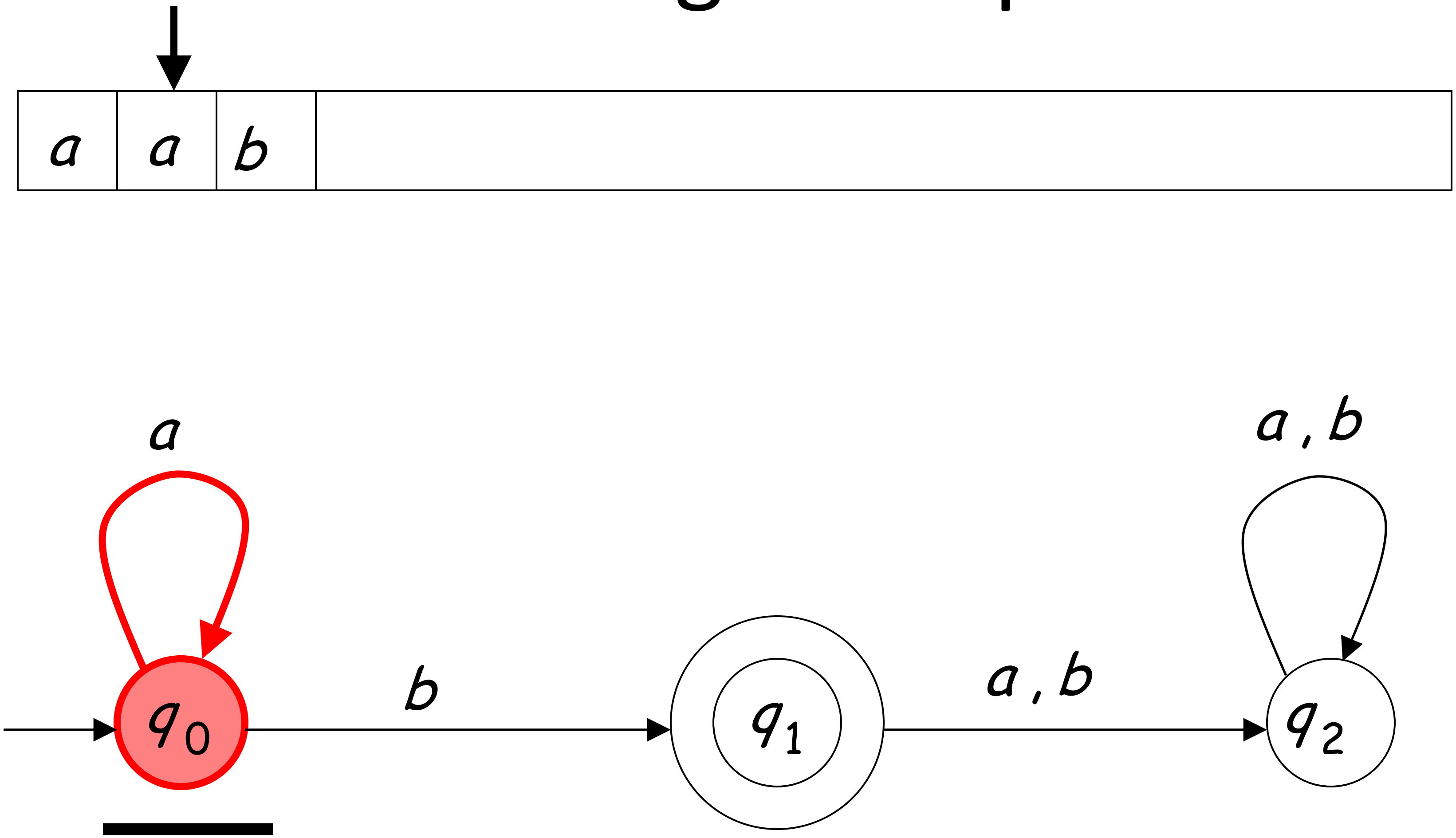




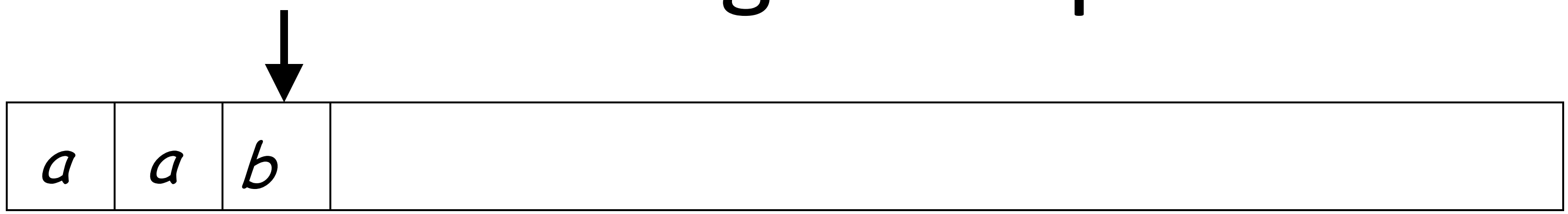
Scanning the Input



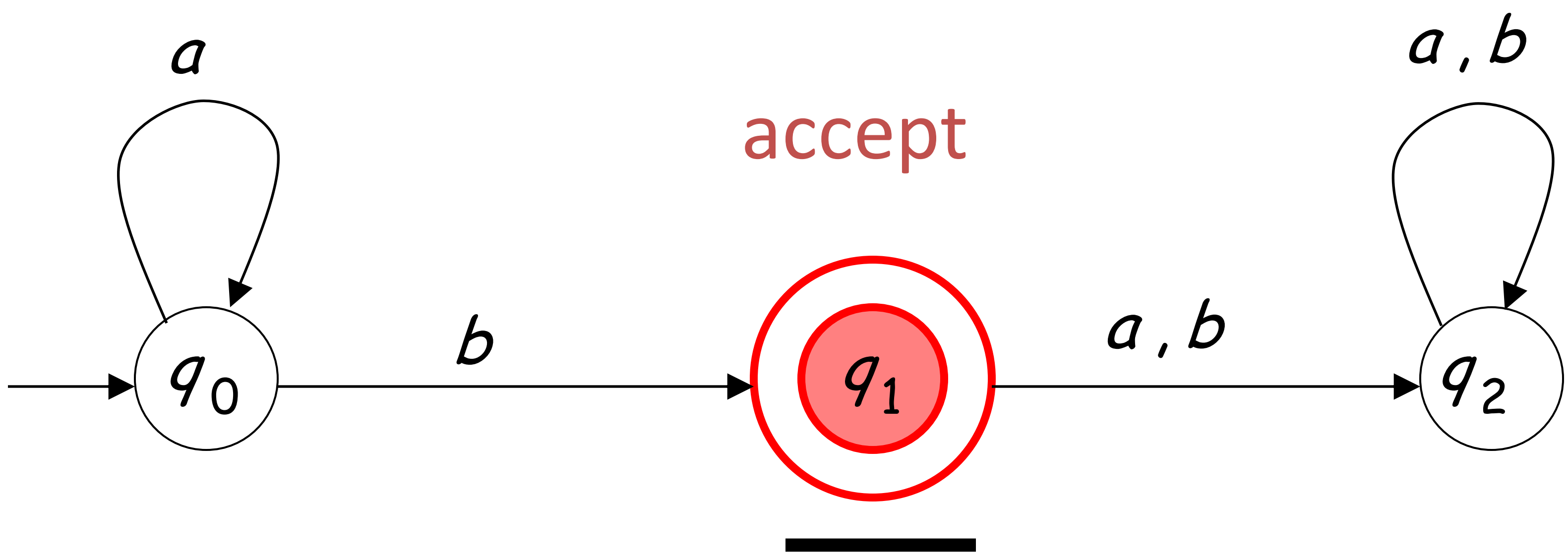
Scanning the Input



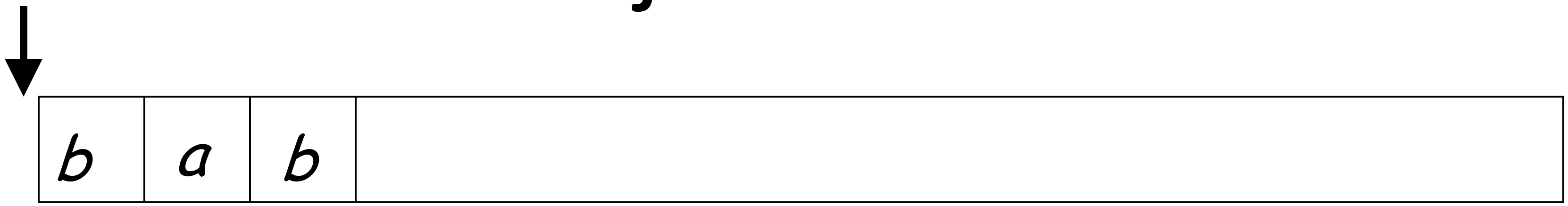
Scanning the Input



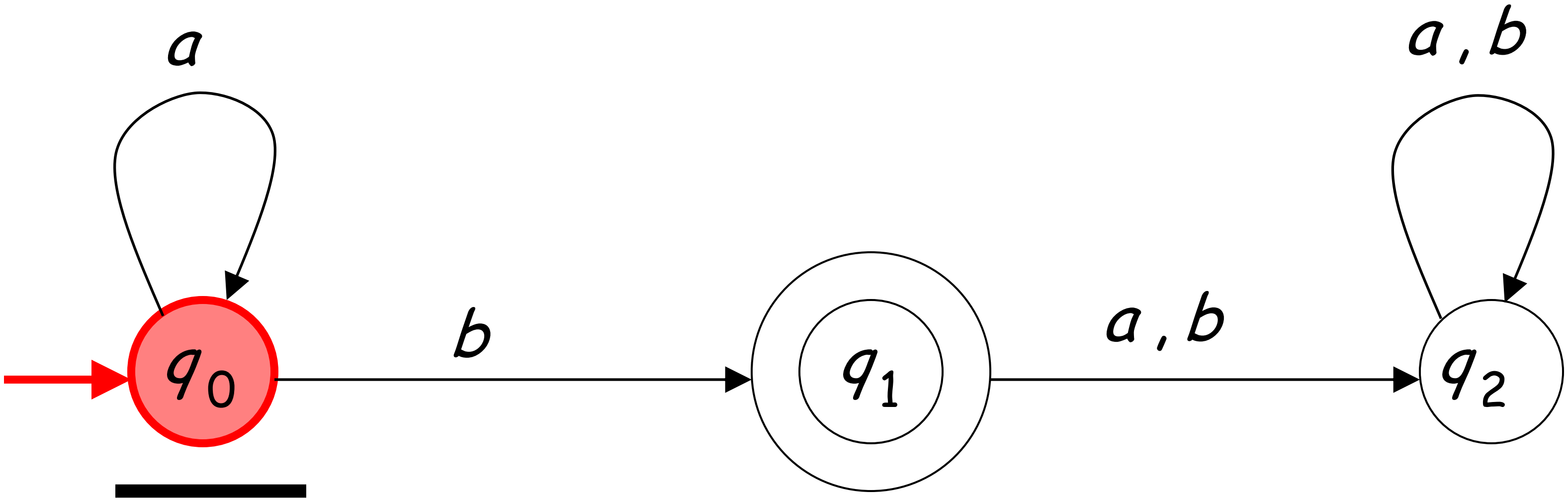
Input finished



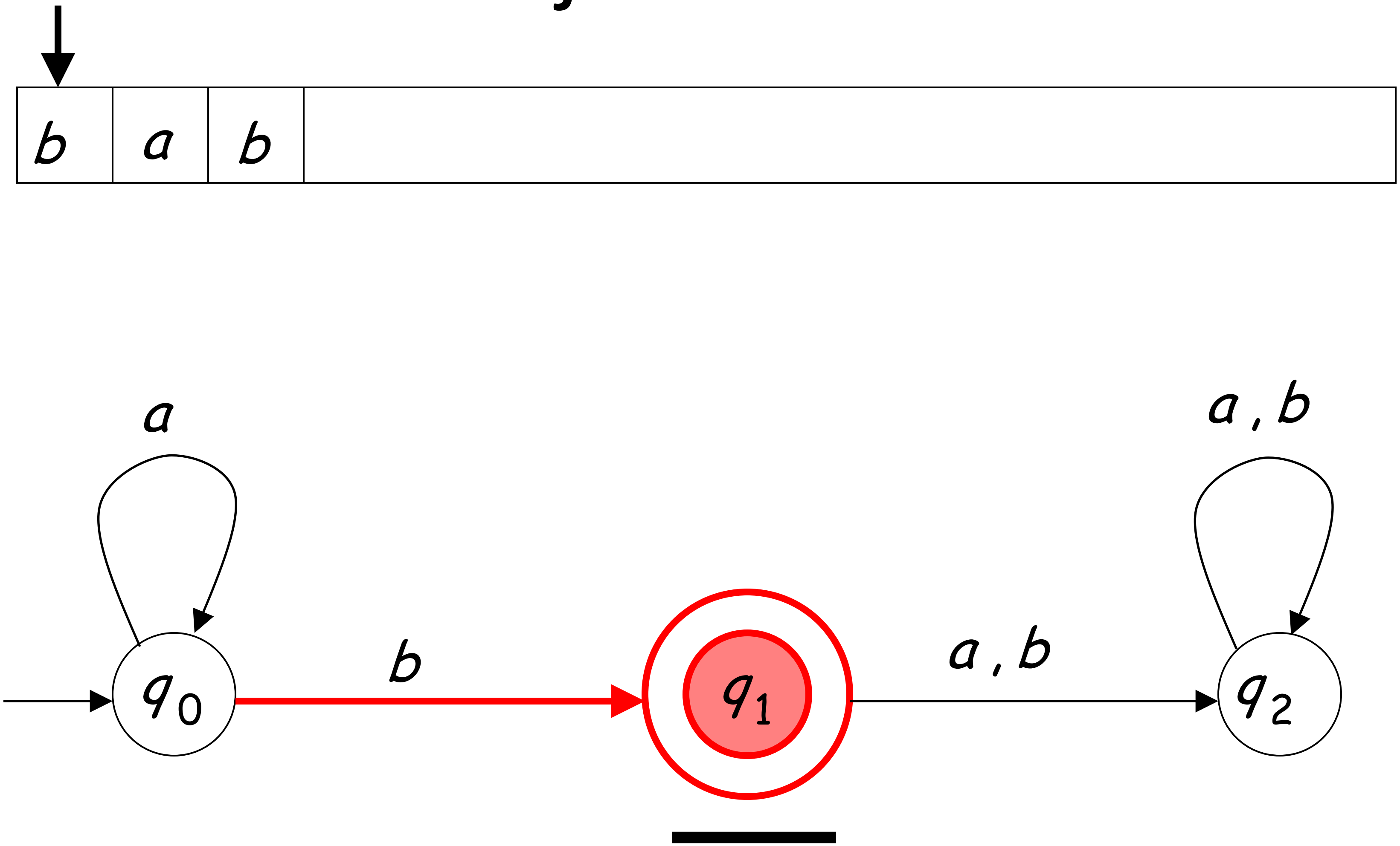
A rejection case



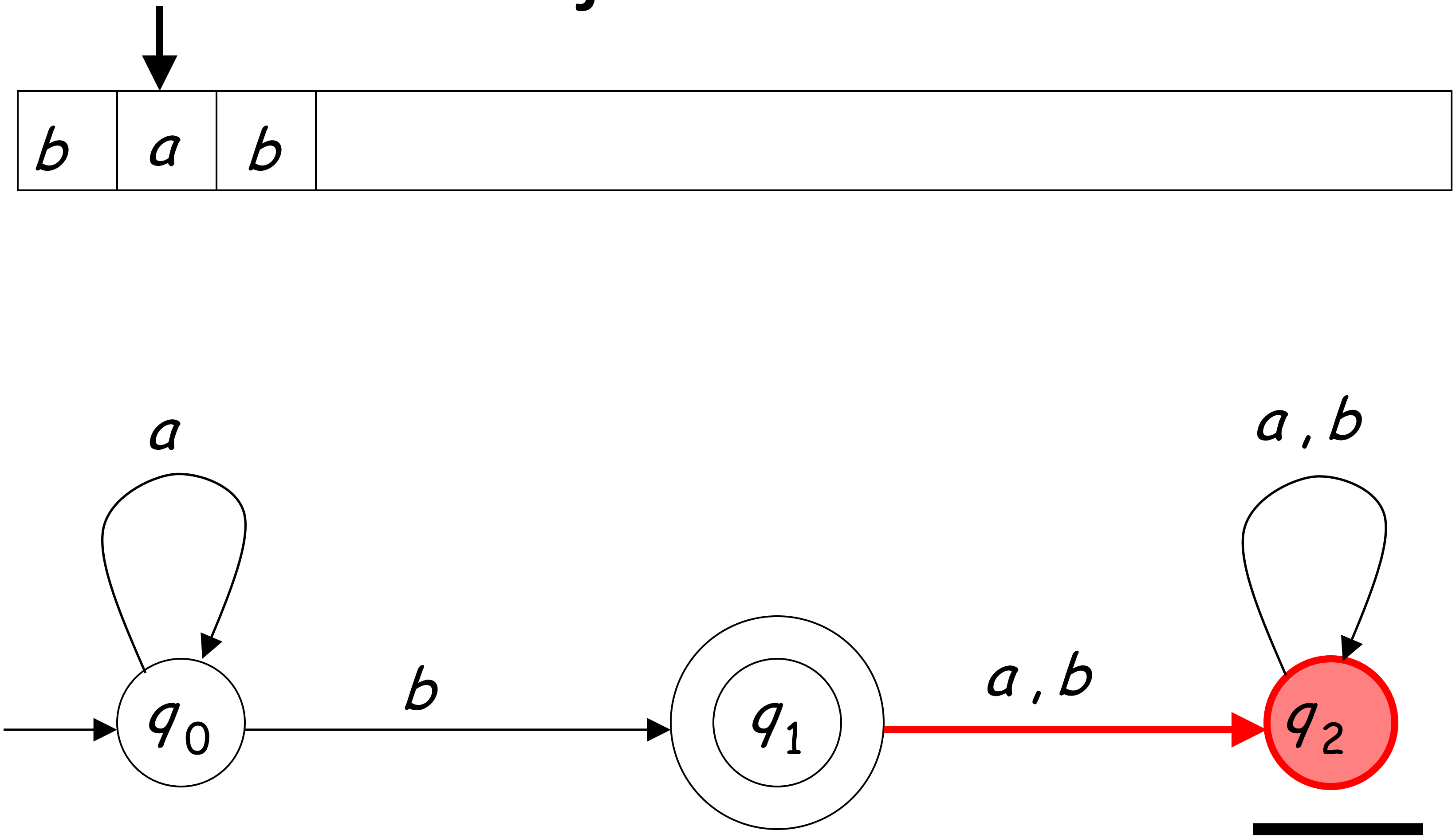
Input String



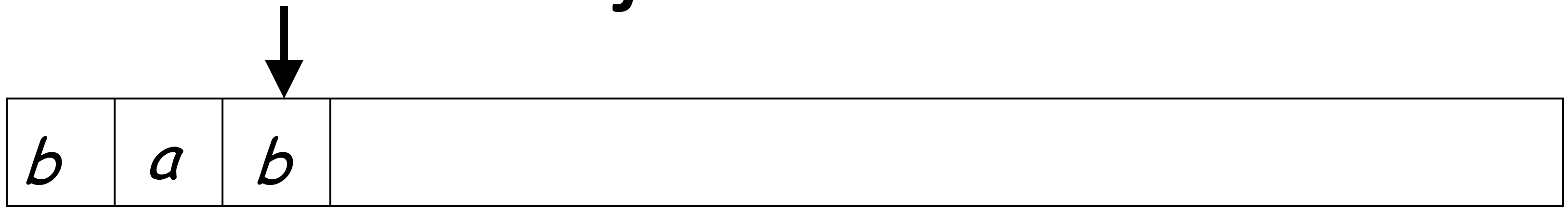
A rejection case



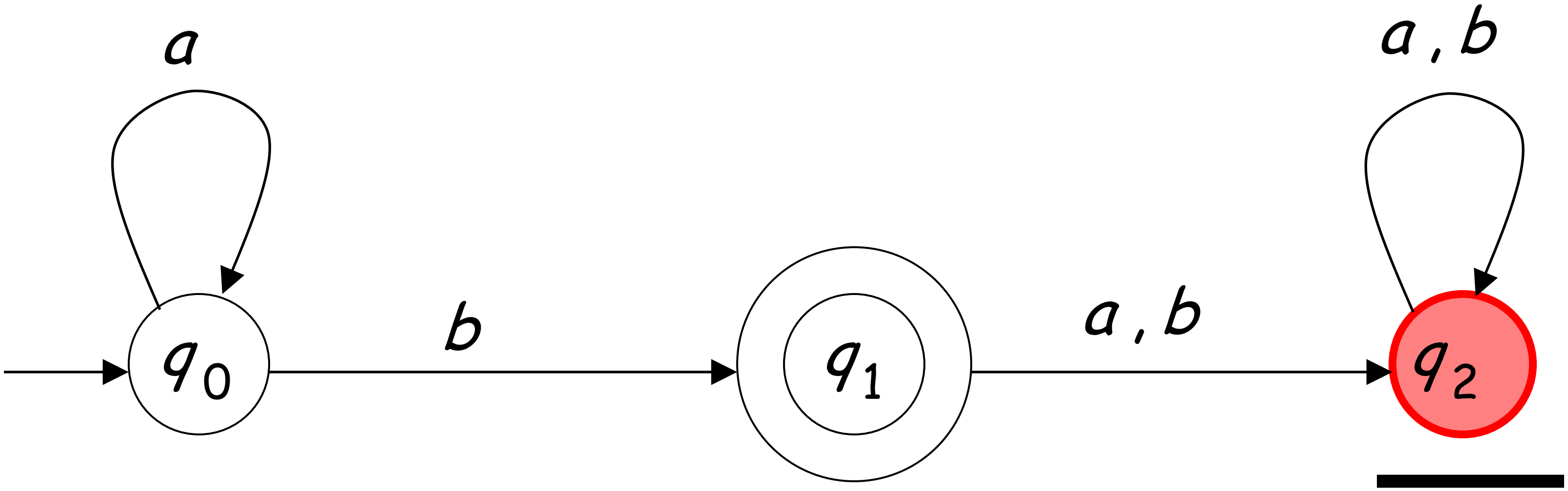
A rejection case



A rejection case



Input finished



reject

Another Example

Language Accepted: $L = \{a^n b : n \geq 0\}$