Theory of Automata & Formal Languages

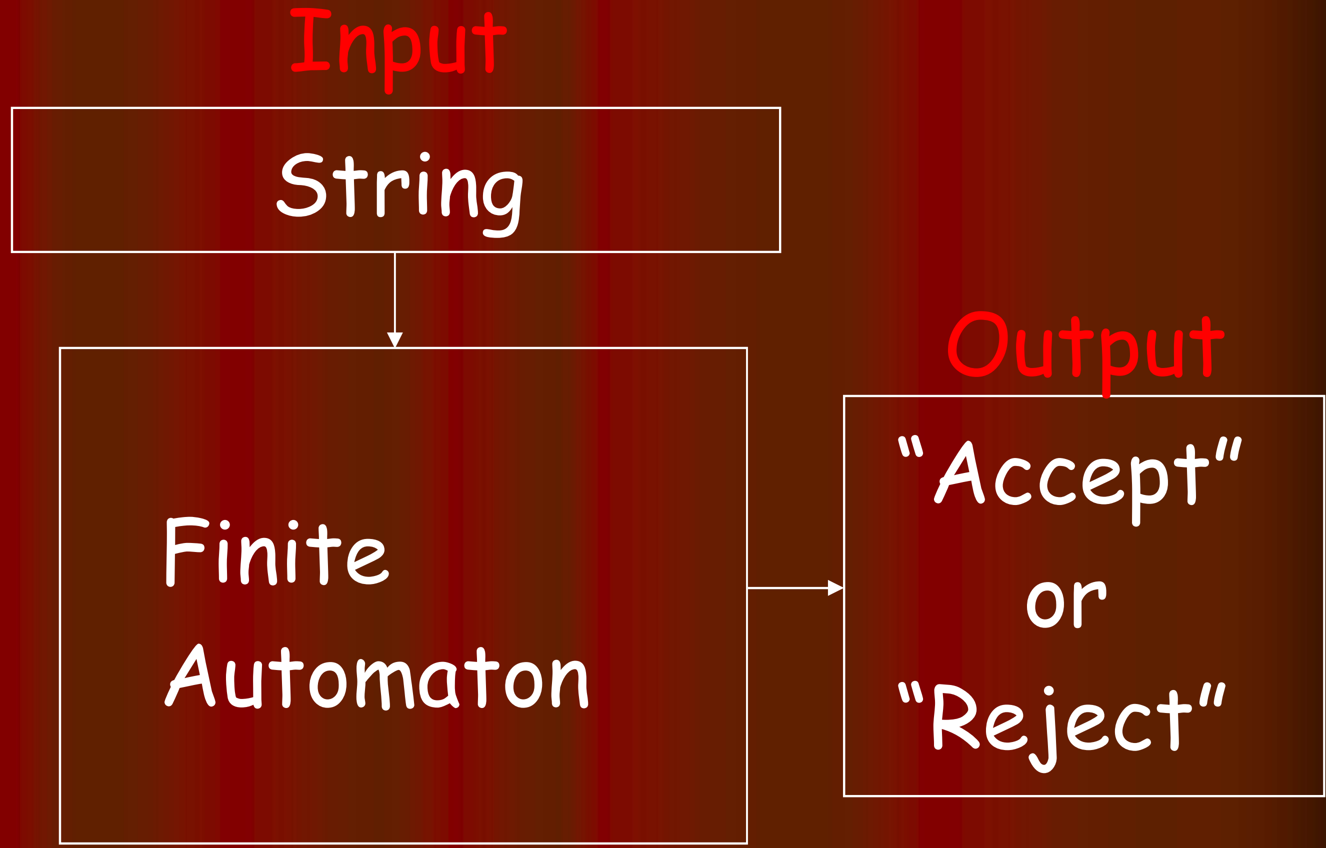
(Theory of Computation)

Compiled By

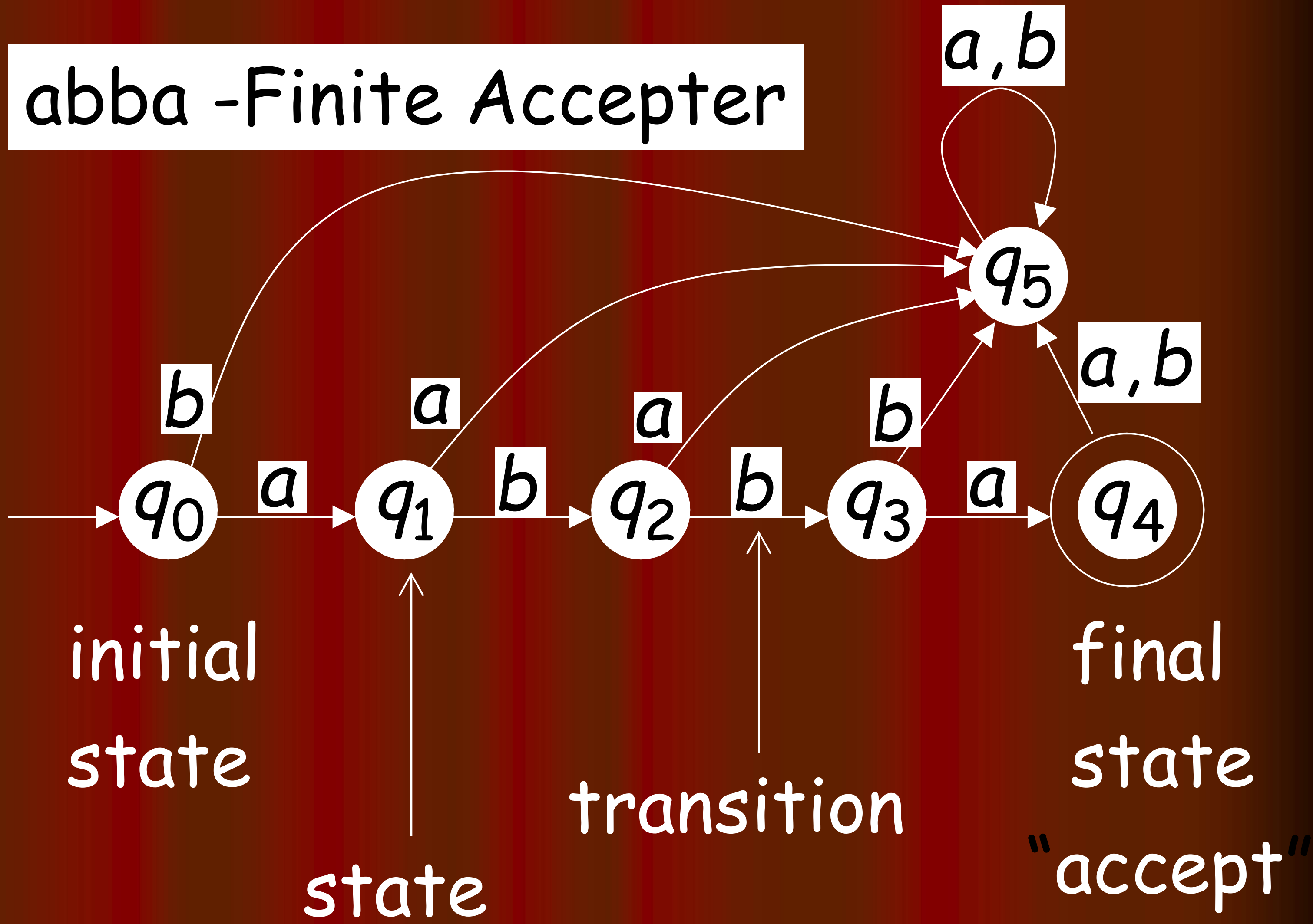
Prof. M. S. Bhatt

Finite Automata

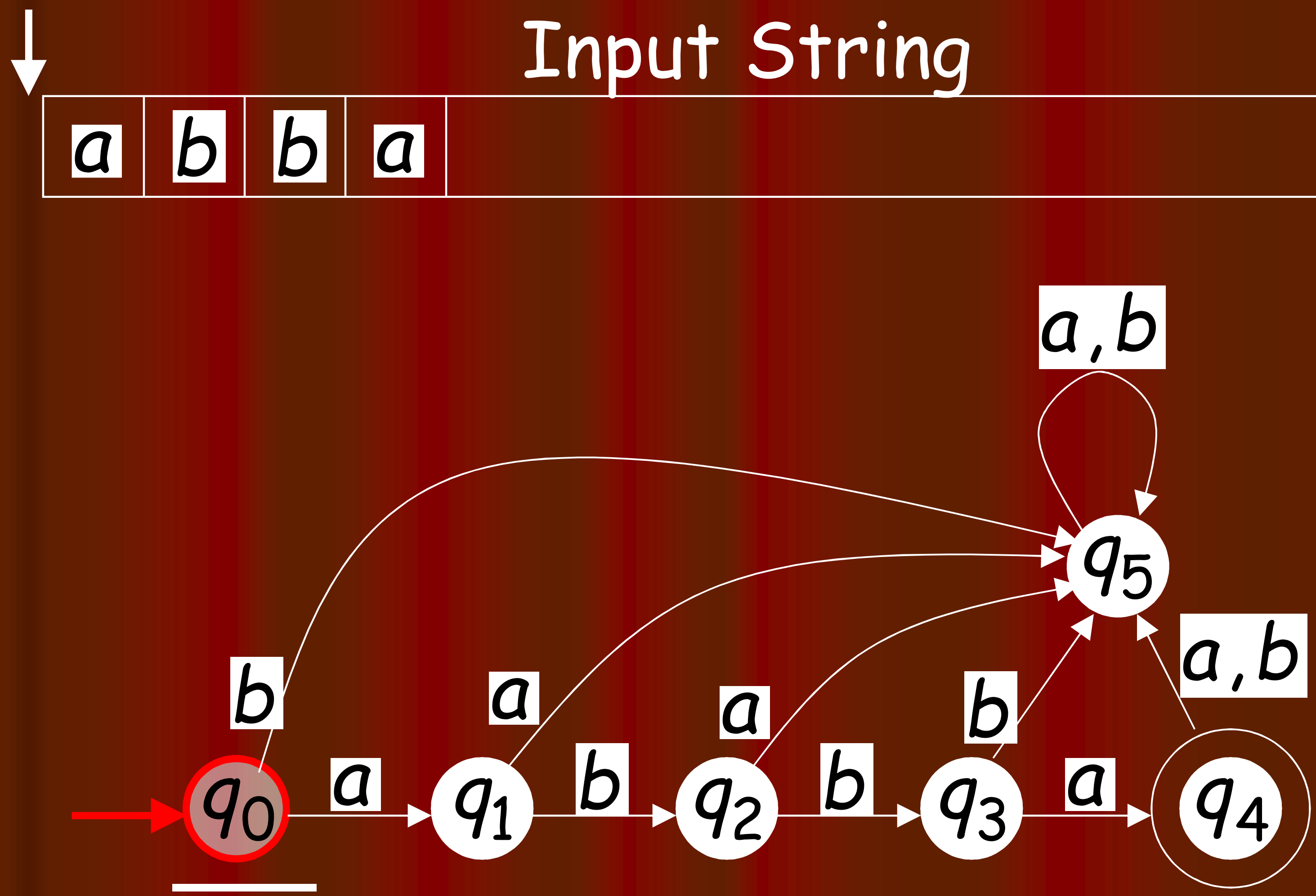
Finite Accepter



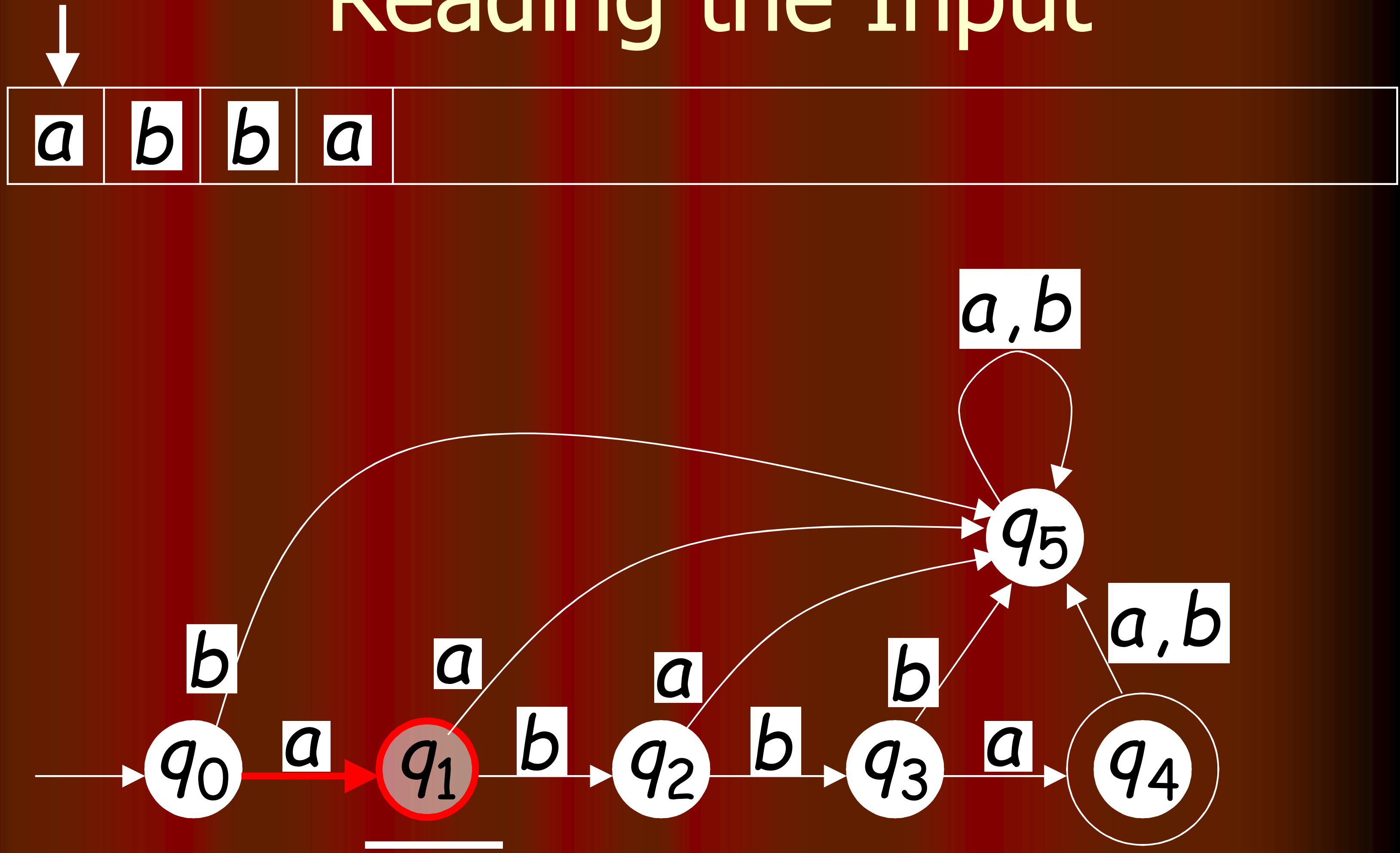
Transition Graph

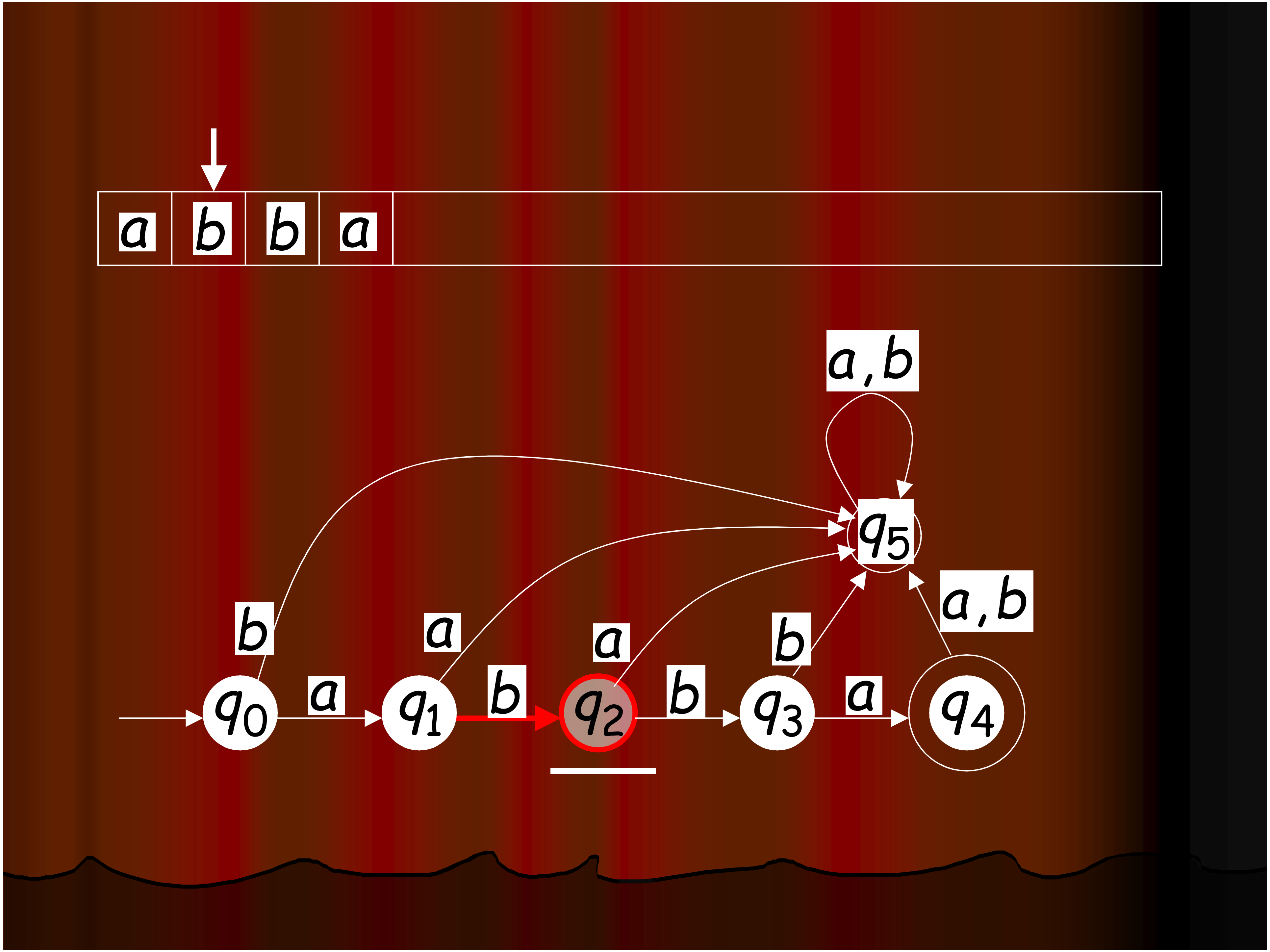


Initial Configuration

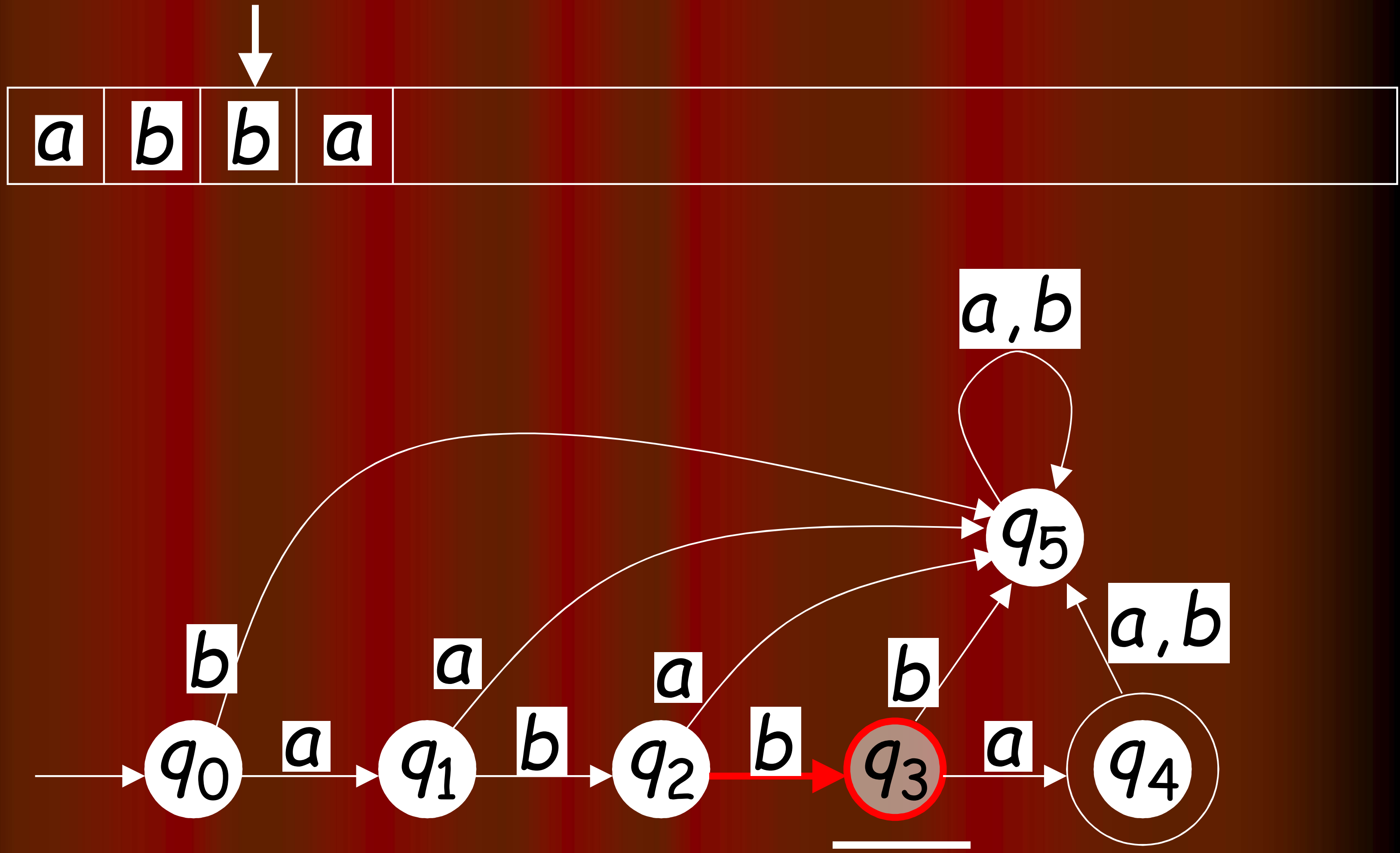


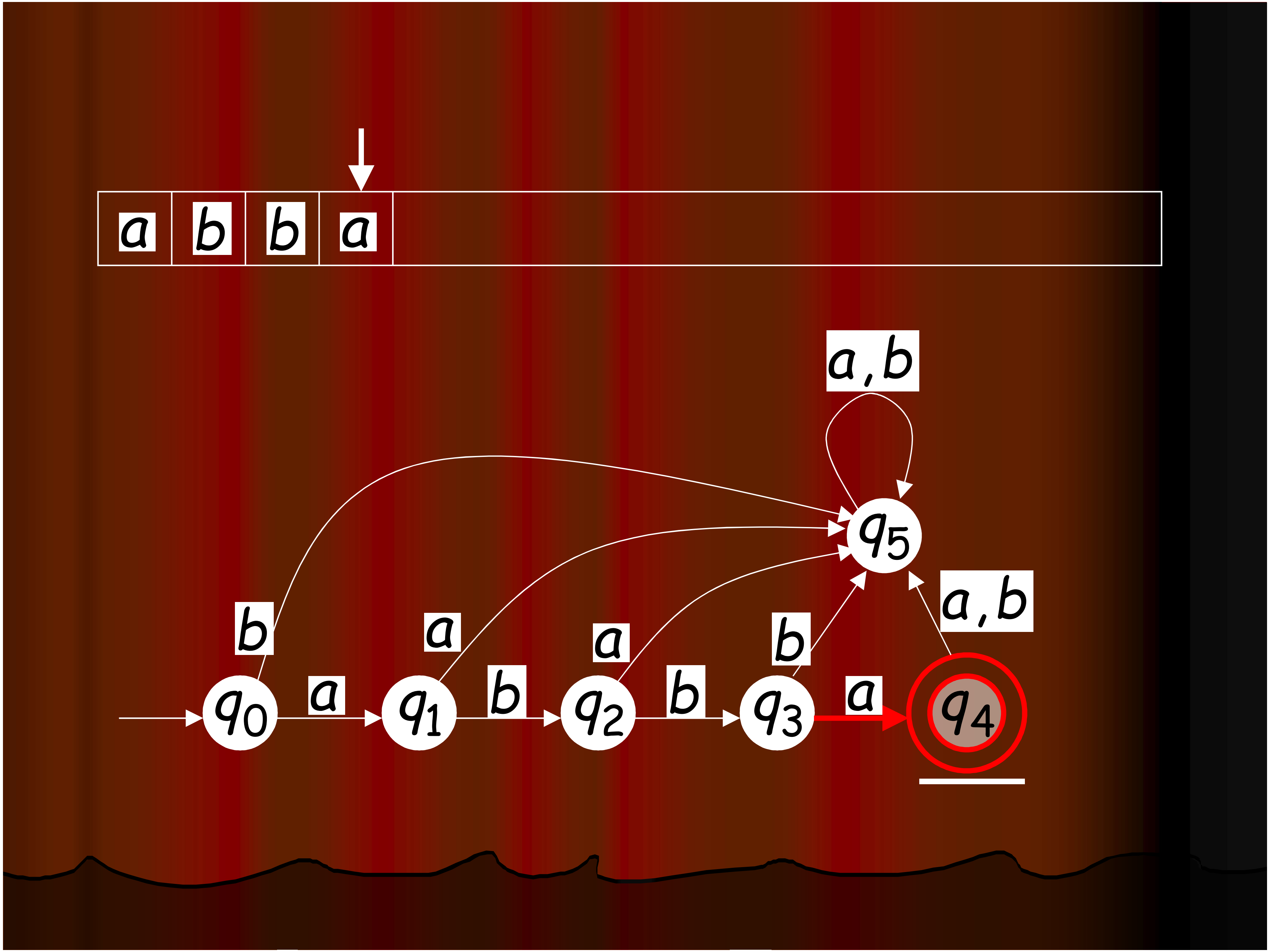
Reading the Input

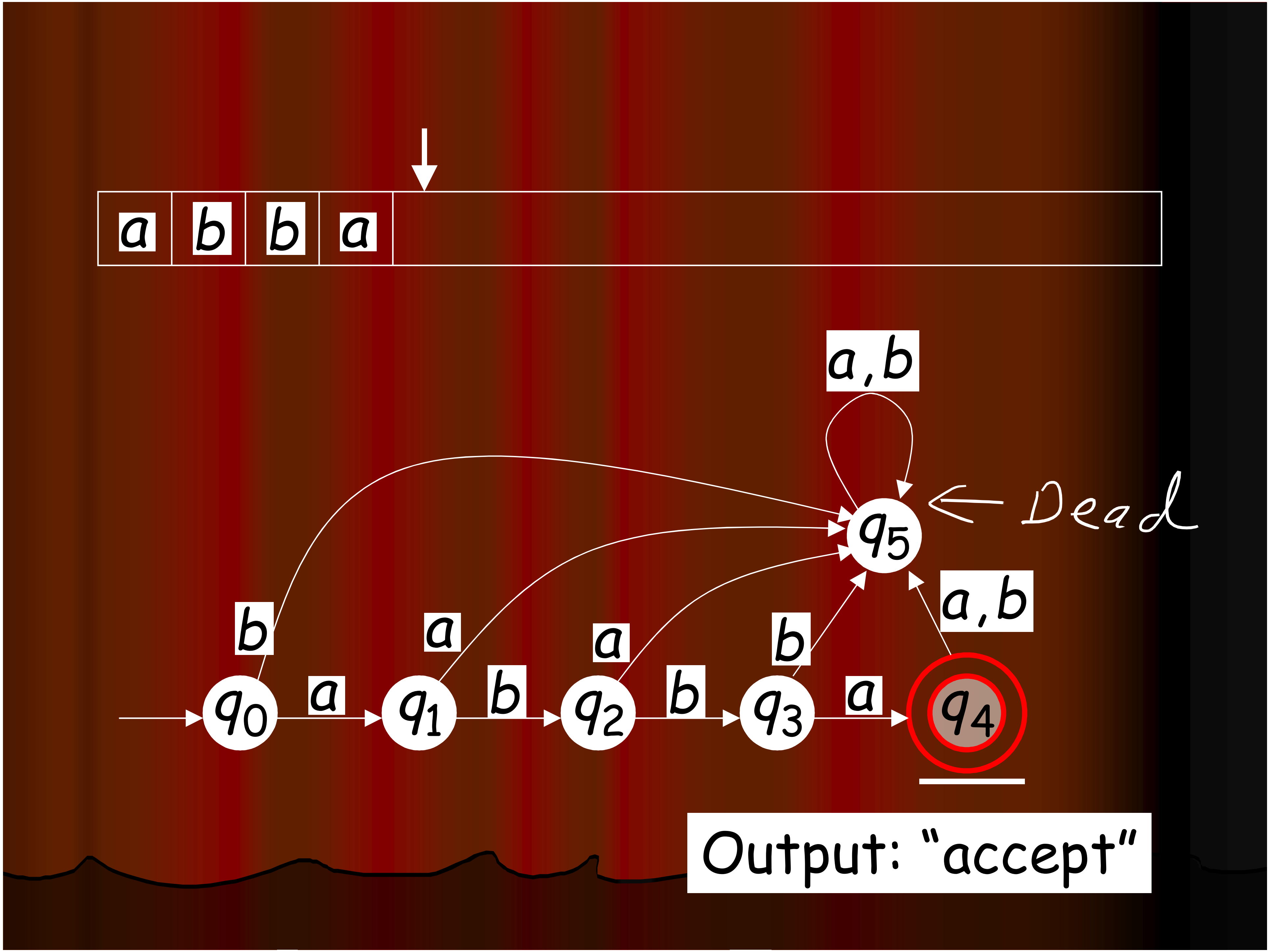




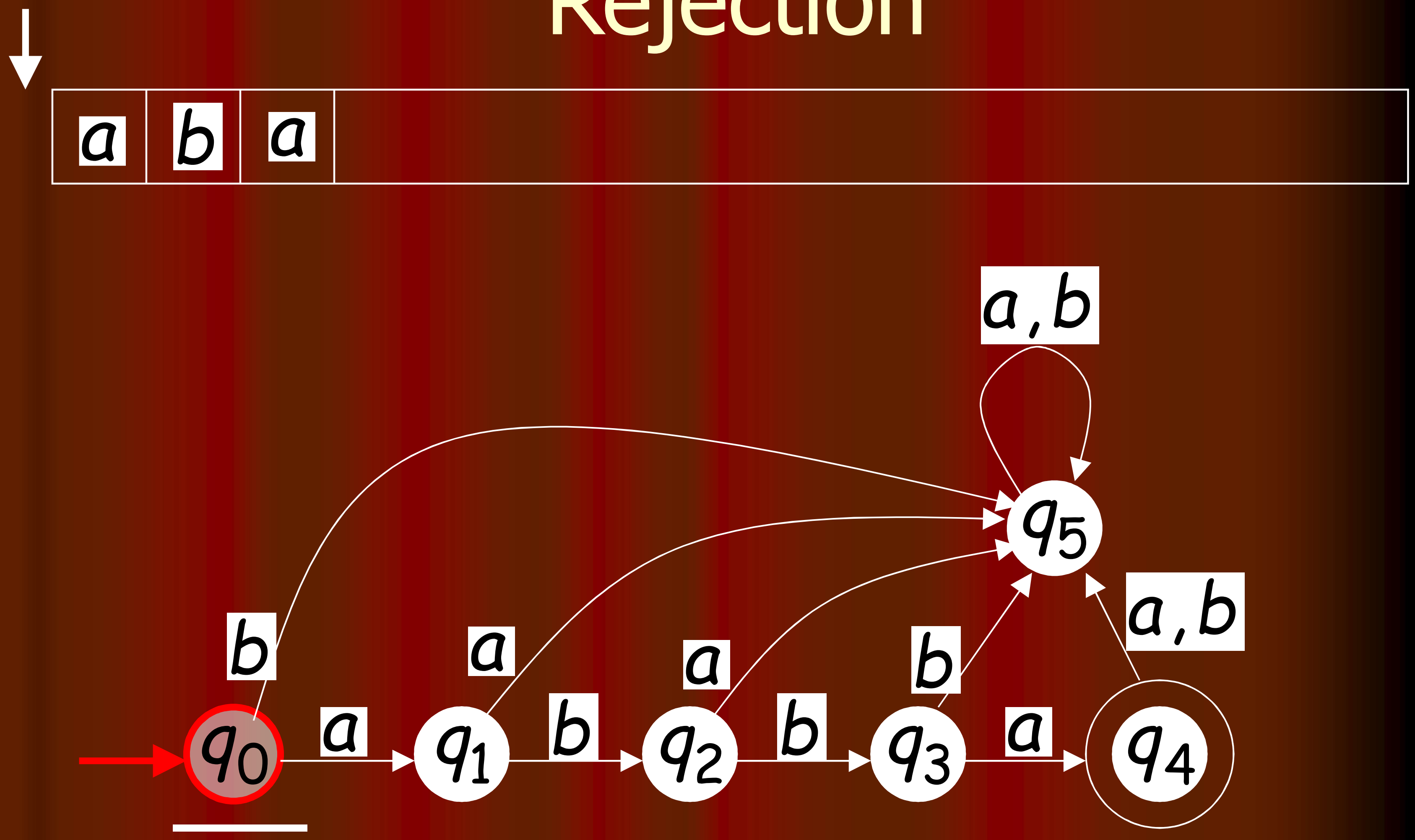
a b b a
a,b
q5
a,b
b
a
a
b
q0
a
q1
b
q2
b
q3
a
q4

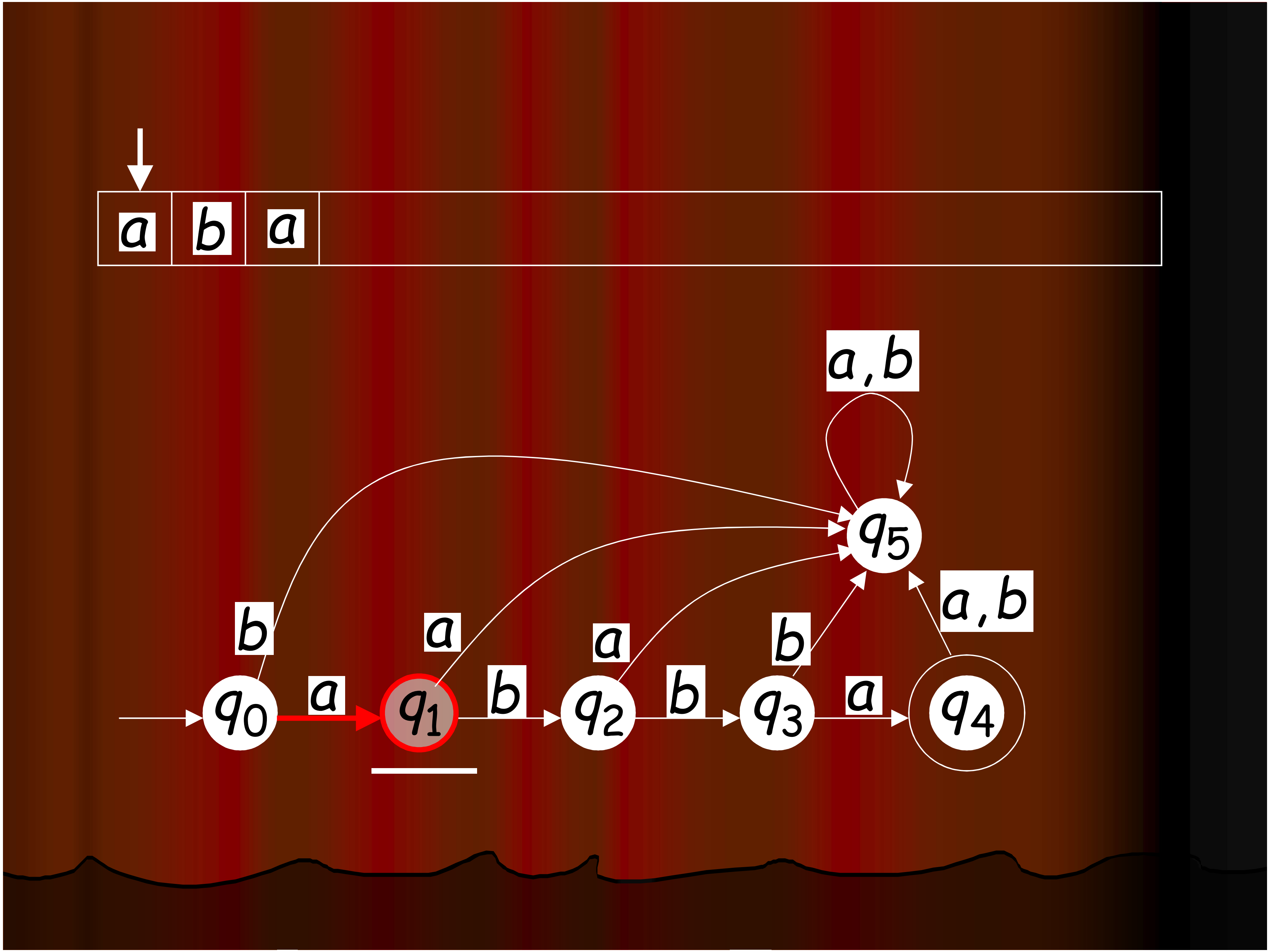

a
b
b
a
a,b
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4

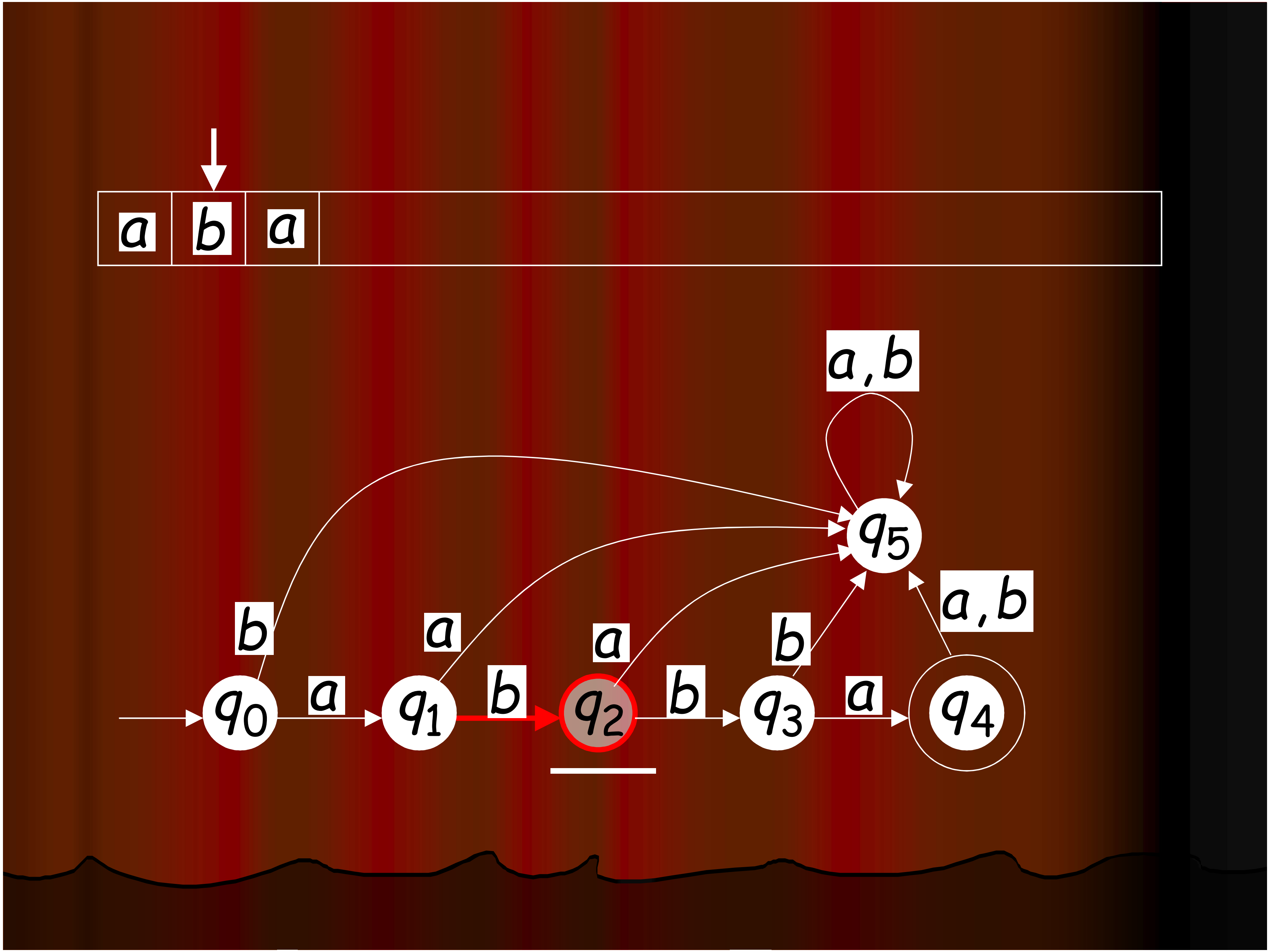

a b b a
a,b
a,b
b
a
a
b
q5
q0
q1
q2
q3
q4
a
b
b
a

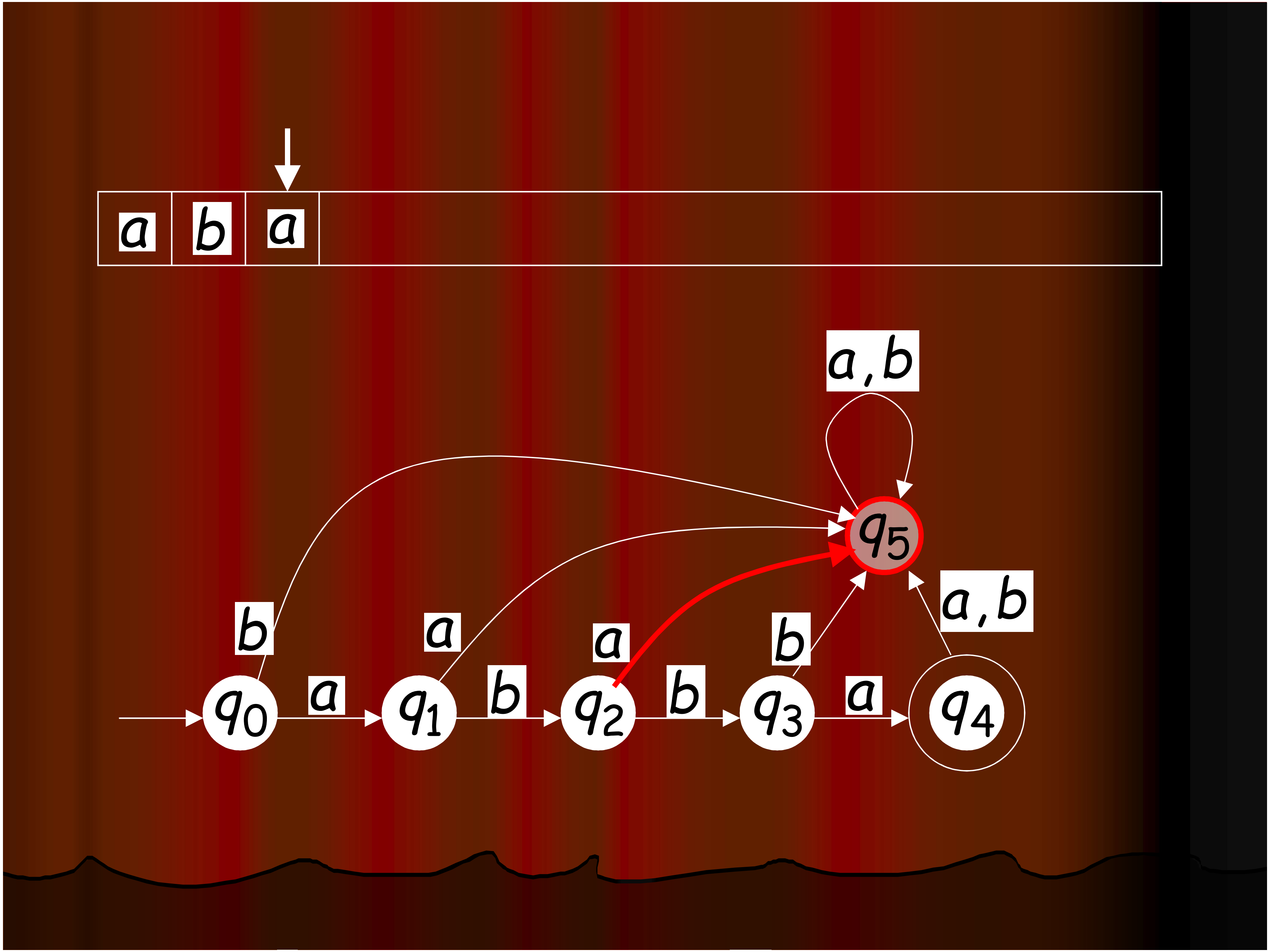

a b b a
a,b
Dead
q5
b
a
a
b
a,b
q0
a
q1
b
q2
b
q3
a
q4
Output: "accept"

Rejection

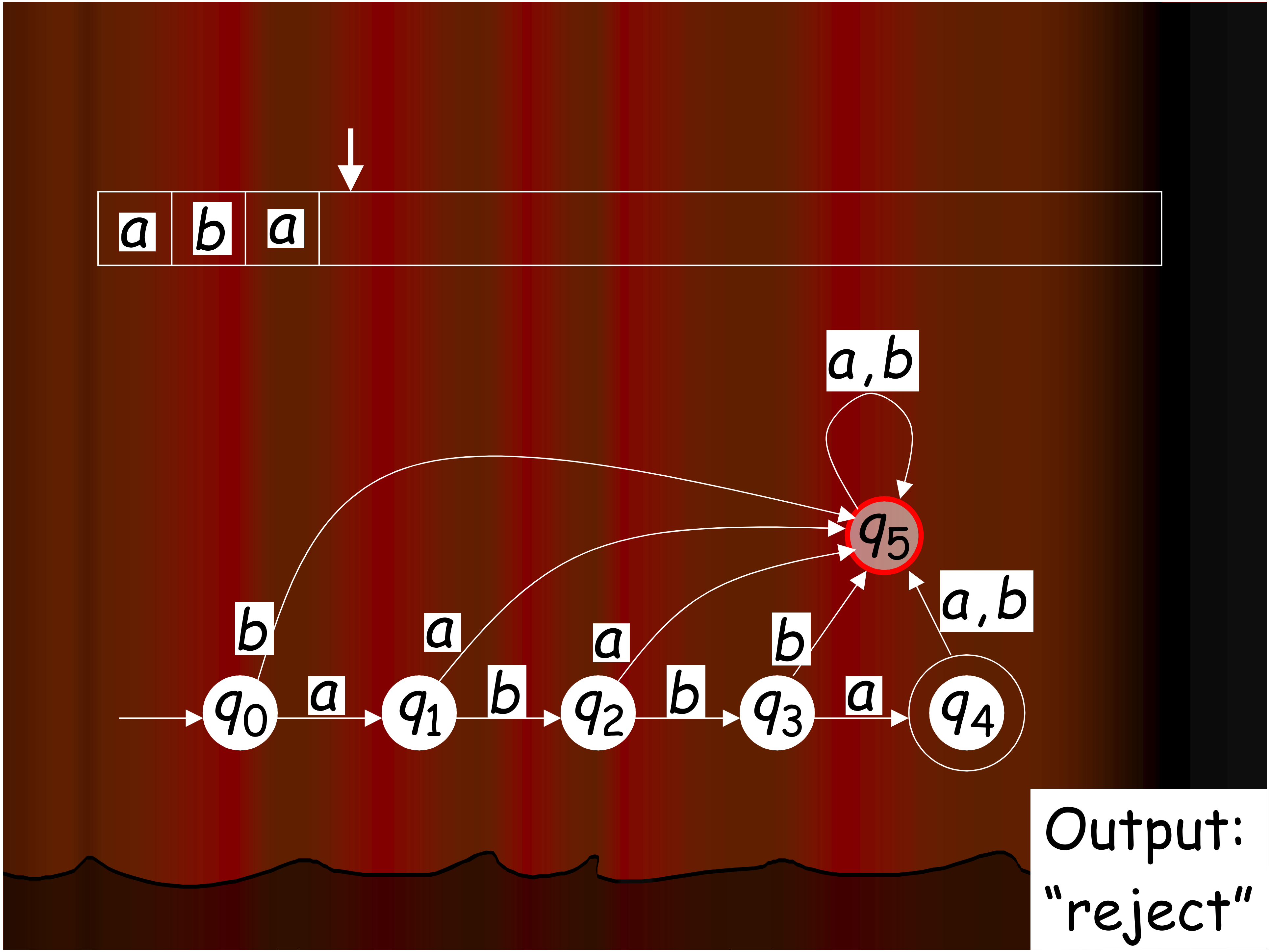



a
b
a
a,b
a,b
b
a
a
b
b
a
a
q_0
q_1
q_2
q_3
q_4
q_5

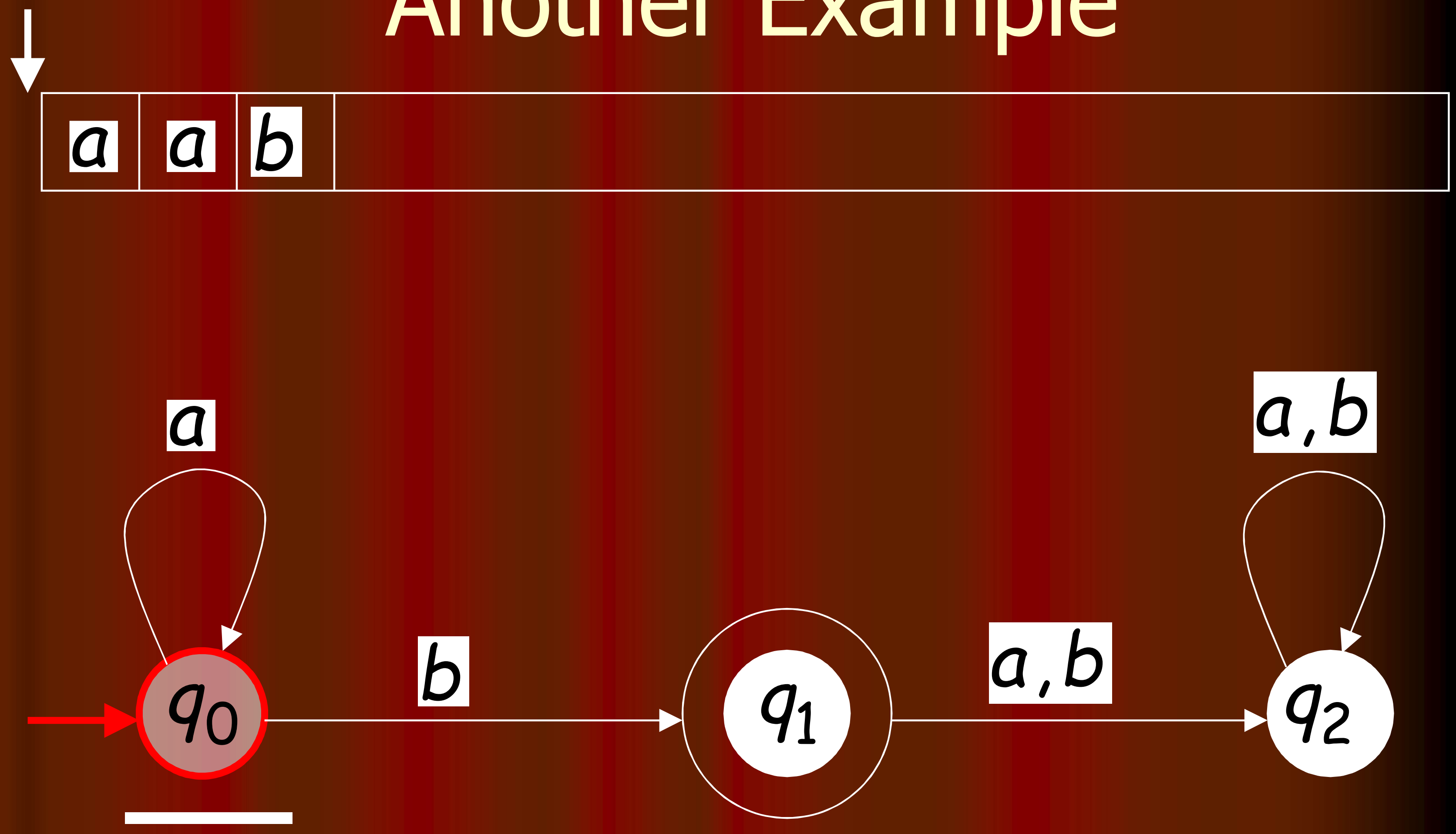

a
b
a
a,b
b
a
a
b
b
a
a,b
q0
q1
q2
q3
q4
q5

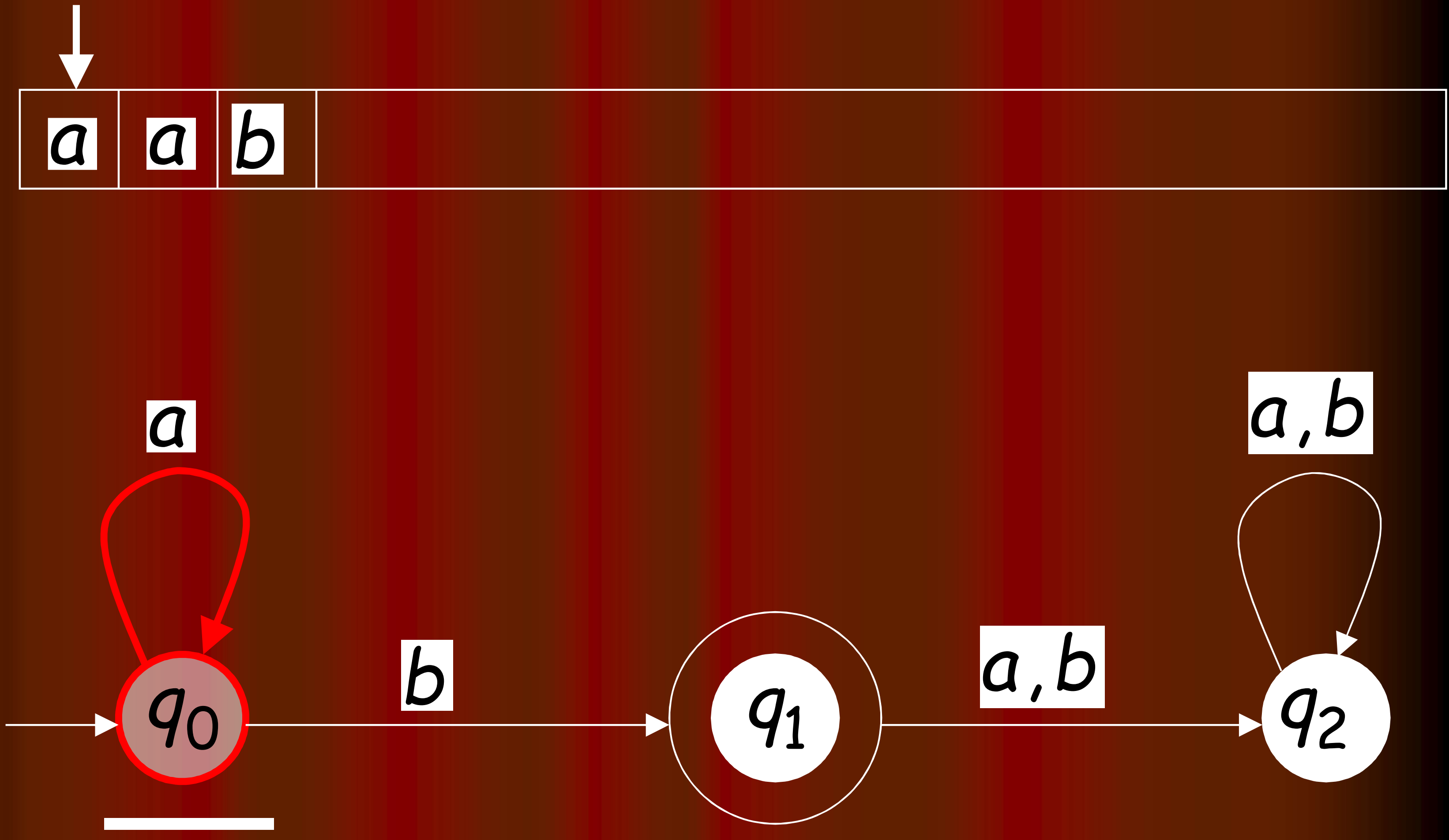


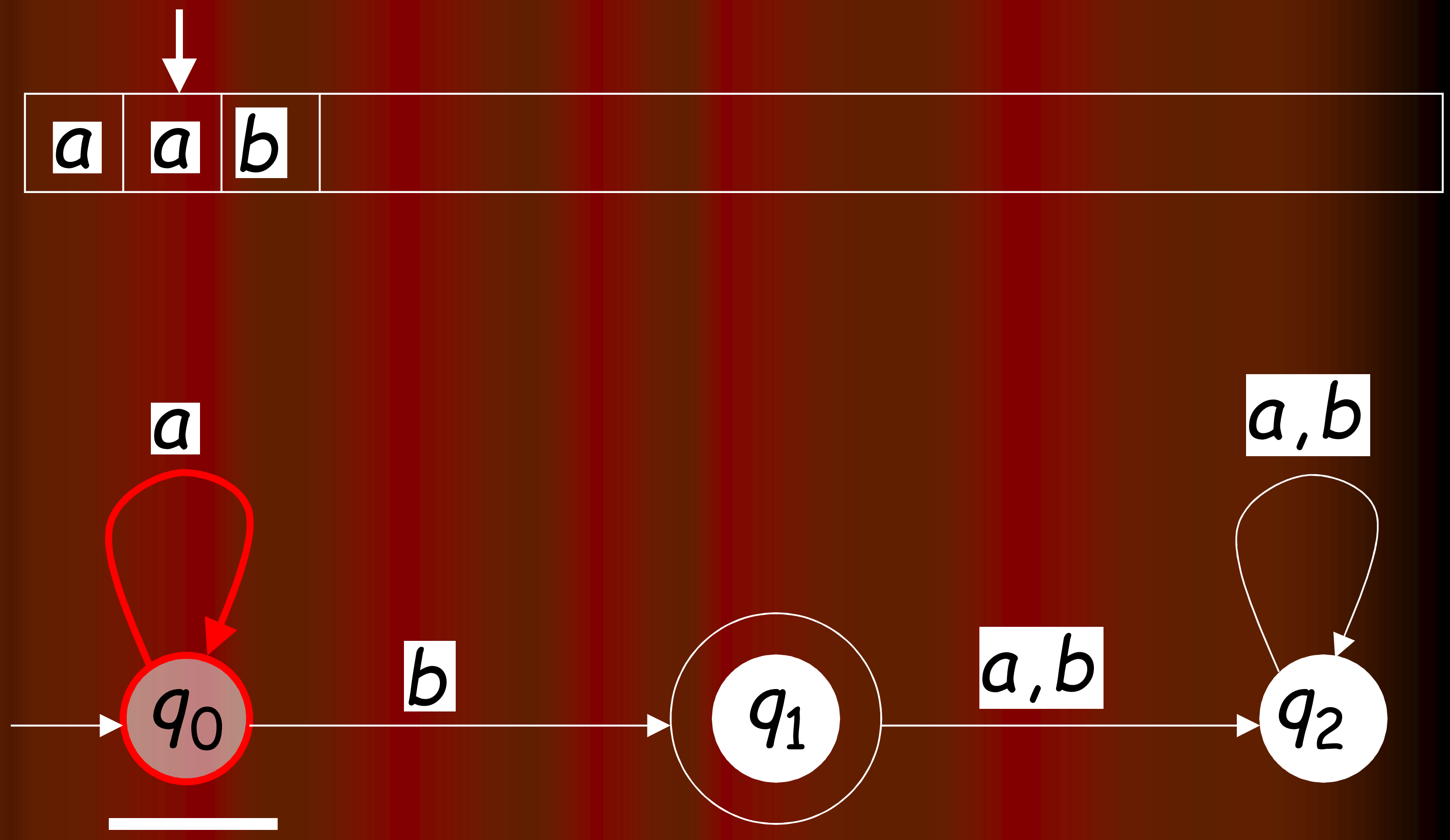
a
b
a
a,b
b
a
a
b
b
a
a,b
q_0
q_1
q_2
q_3
q_4
q_5


a
b
a
a,b
b
a
a
b
a
b
b
a
a,b
q0 q1 q2 q3 q4 q5
Output: "reject"

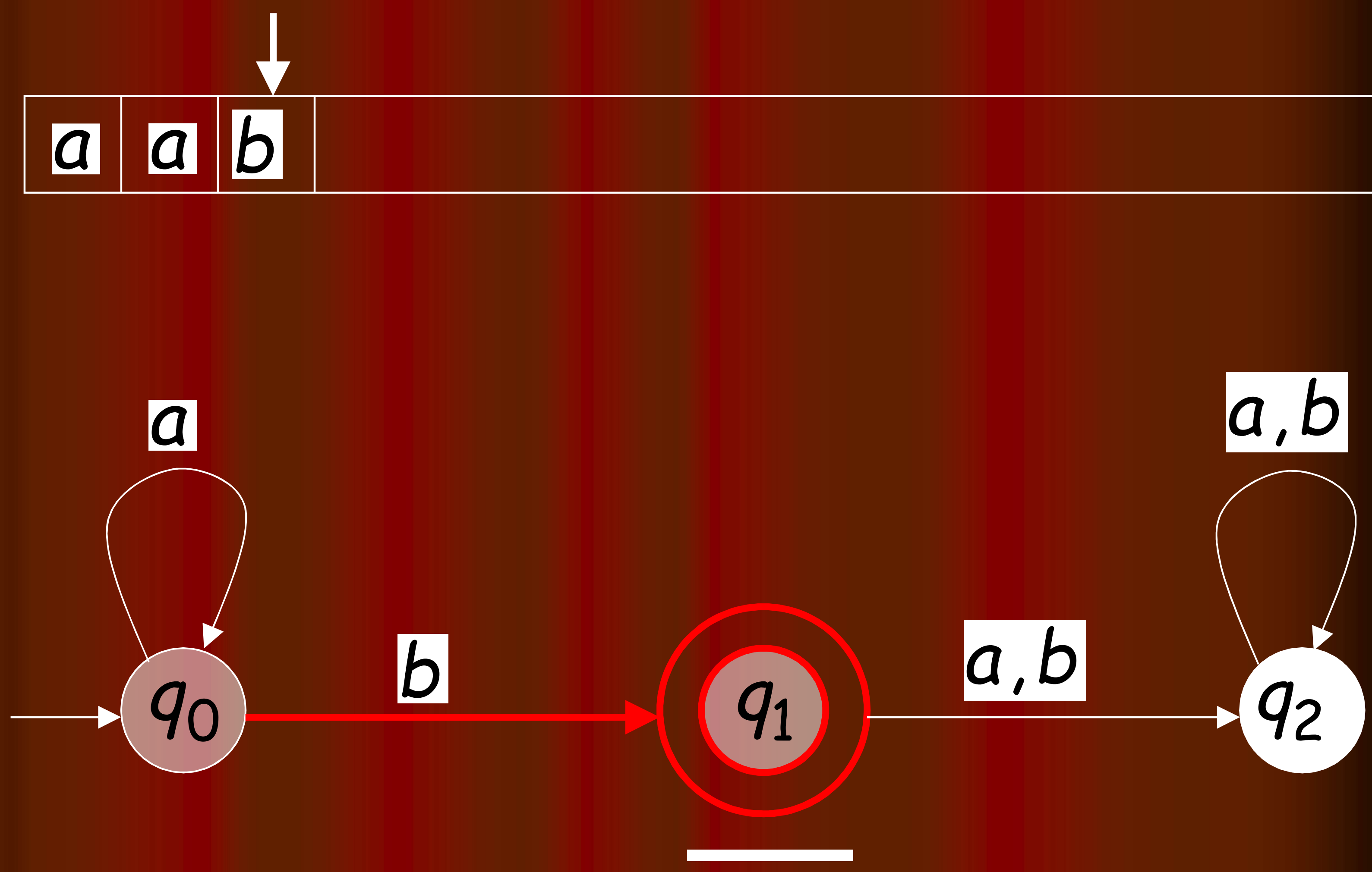
Another Example



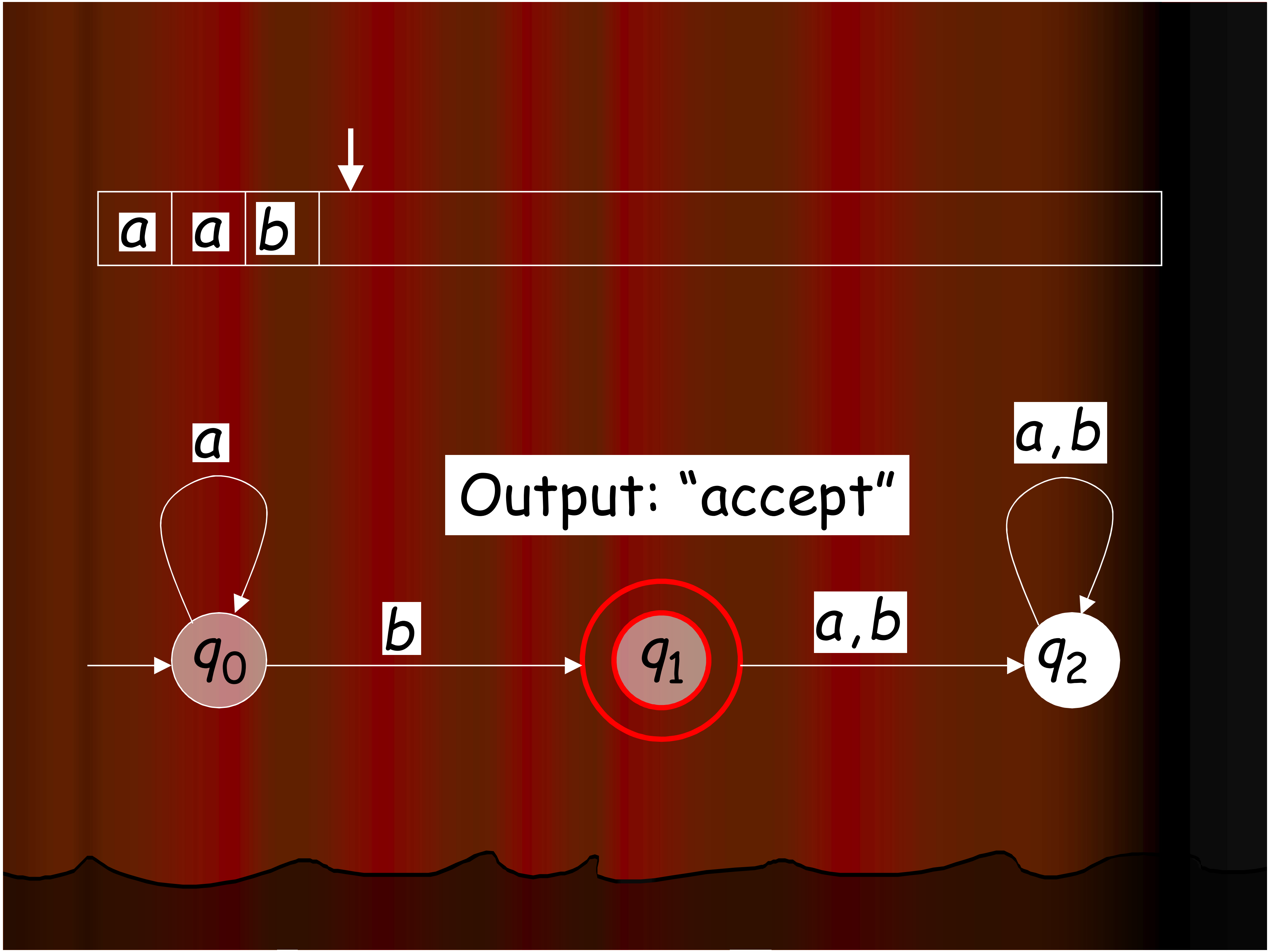

a a b
a
b
a,b
a,b
q_0
q_1
q_2



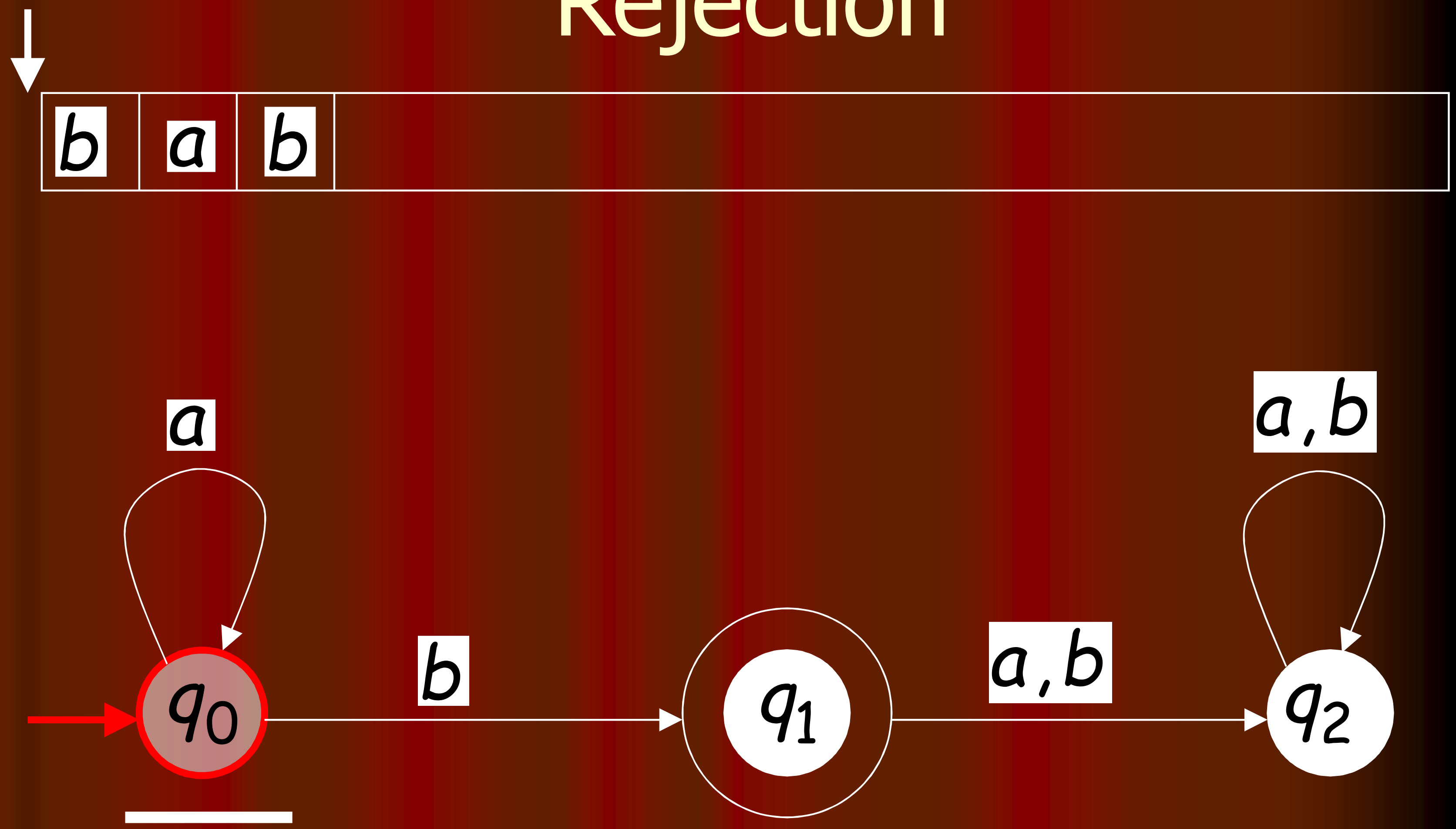
a
a
b
a
b
a,b
a,b
q_0
q_1
q_2

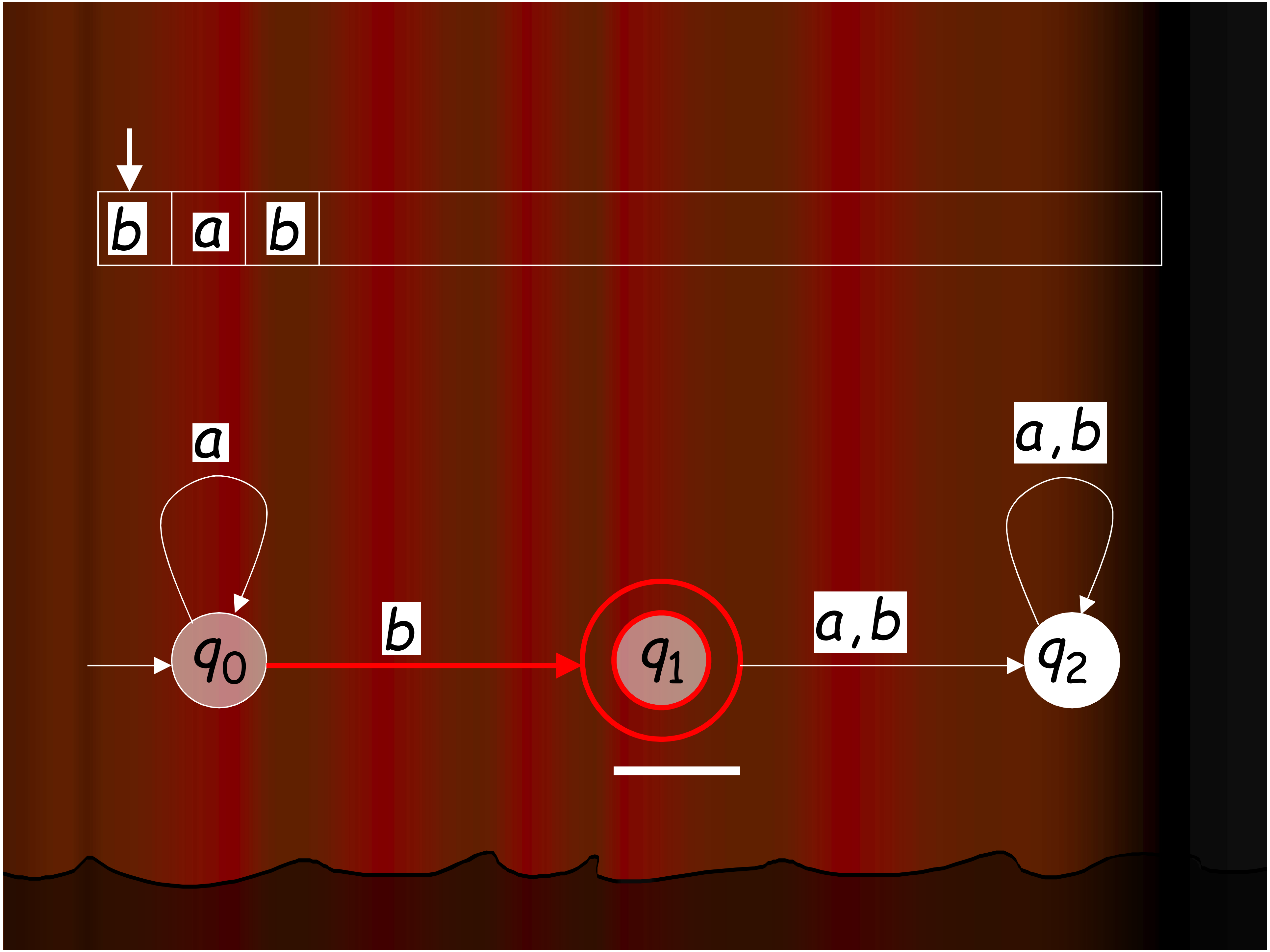


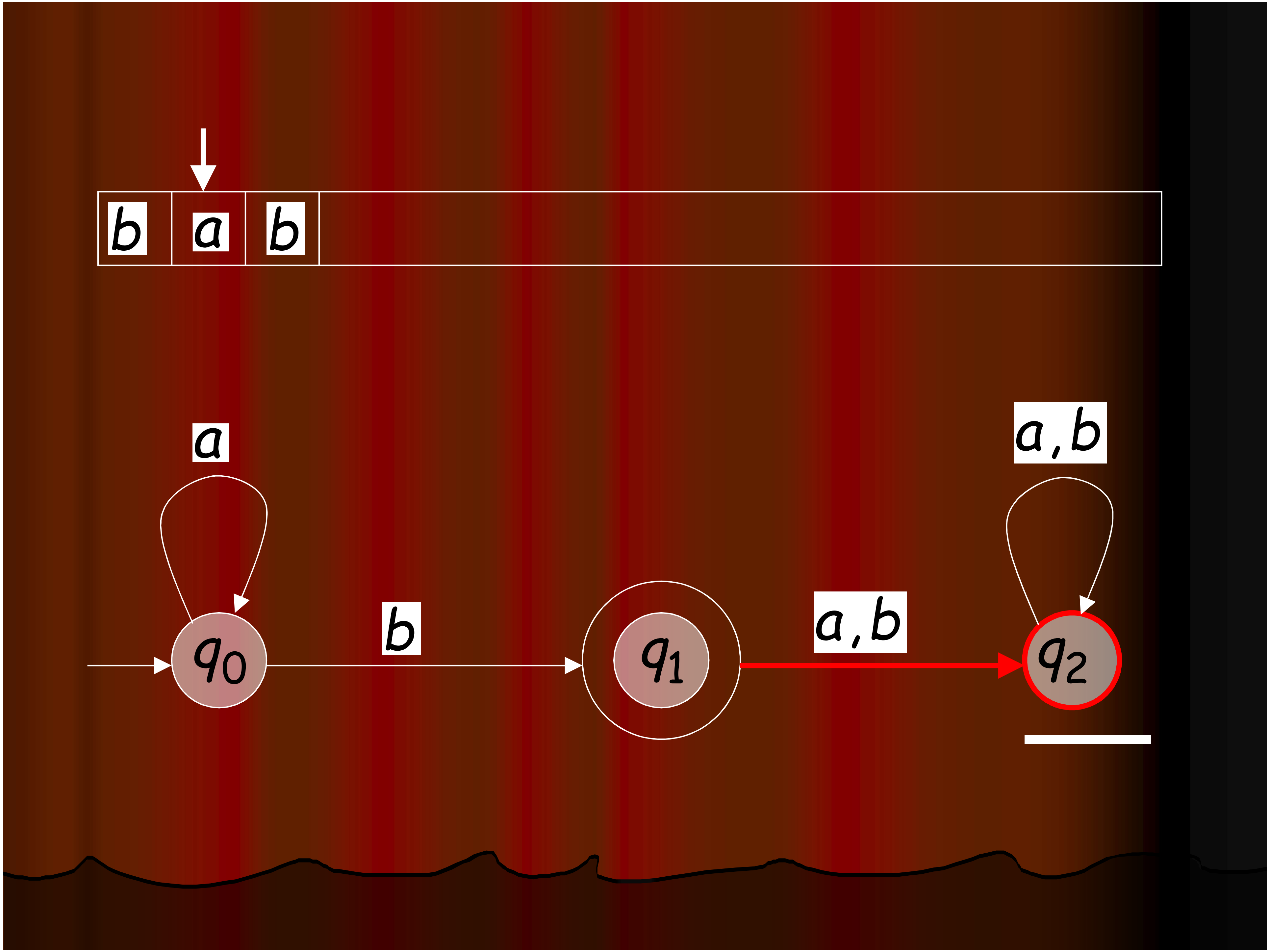
$L = \{a^n b \mid n \geq 0\}$

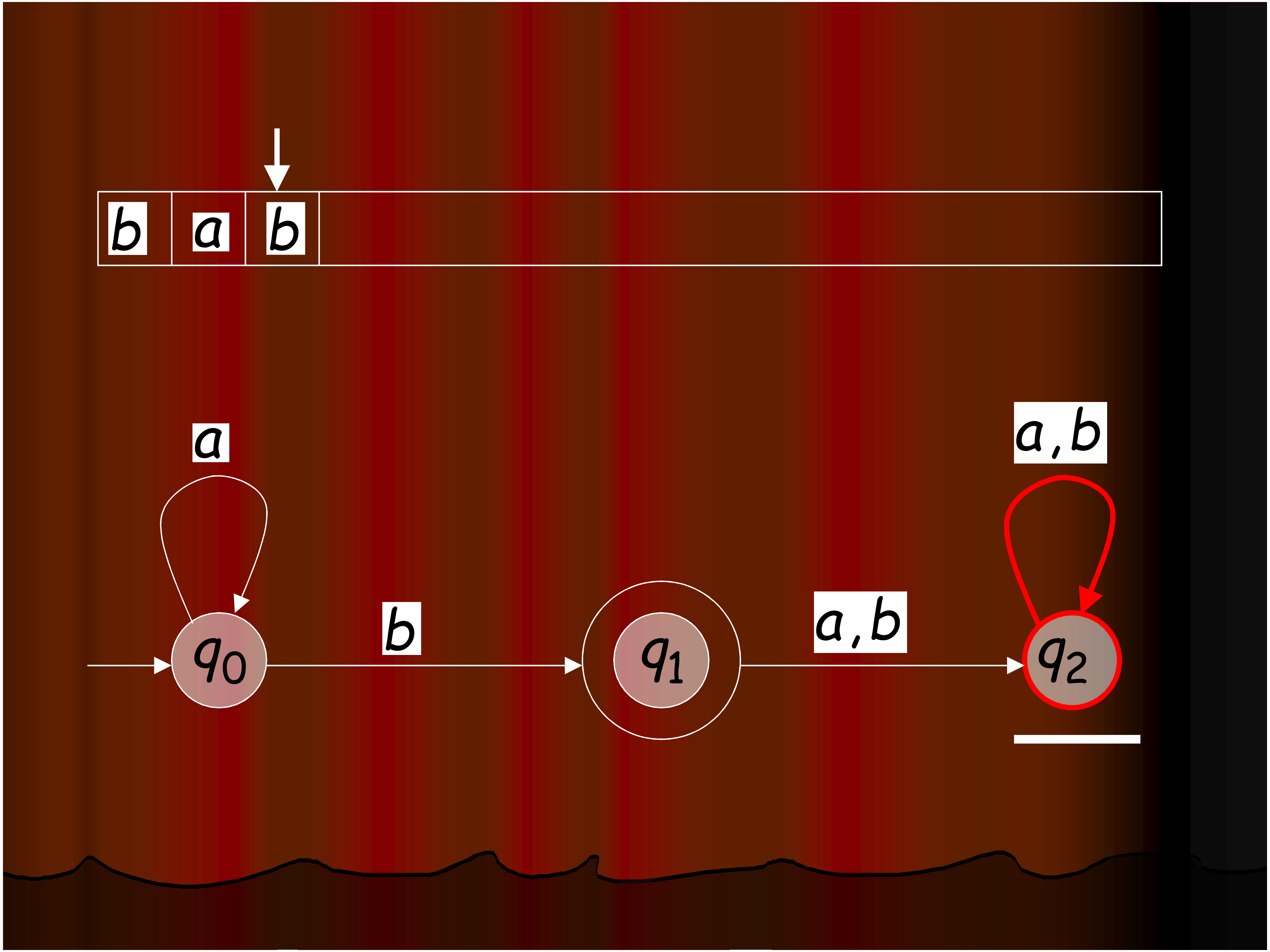

a
a
b
a
Output: "accept"
a,b
b
q_0
q_1
a,b
q_2

Rejection

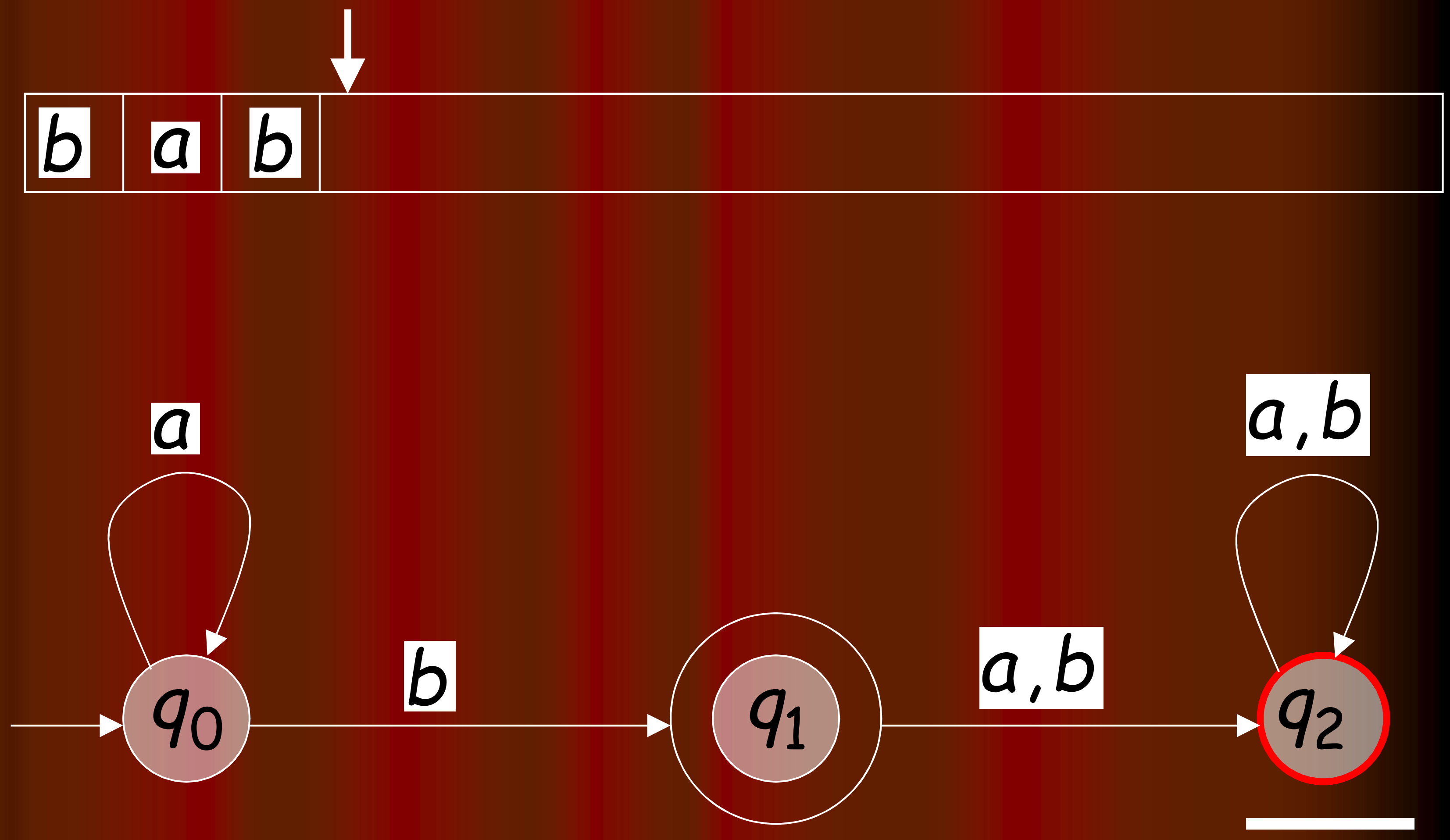



b
a
b
a
b
a,b
a,b
q_0
q_1
q_2


b
a
b
a
b
a,b
a,b
q0
q1
q2



b
a
b
a
b
a,b
a,b
q_0
q_1
q_2



Output: "reject"

Formalities

- Deterministic Finite Accepter (DFA)

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

5-Tuple

Q : set of states

Σ : input alphabet

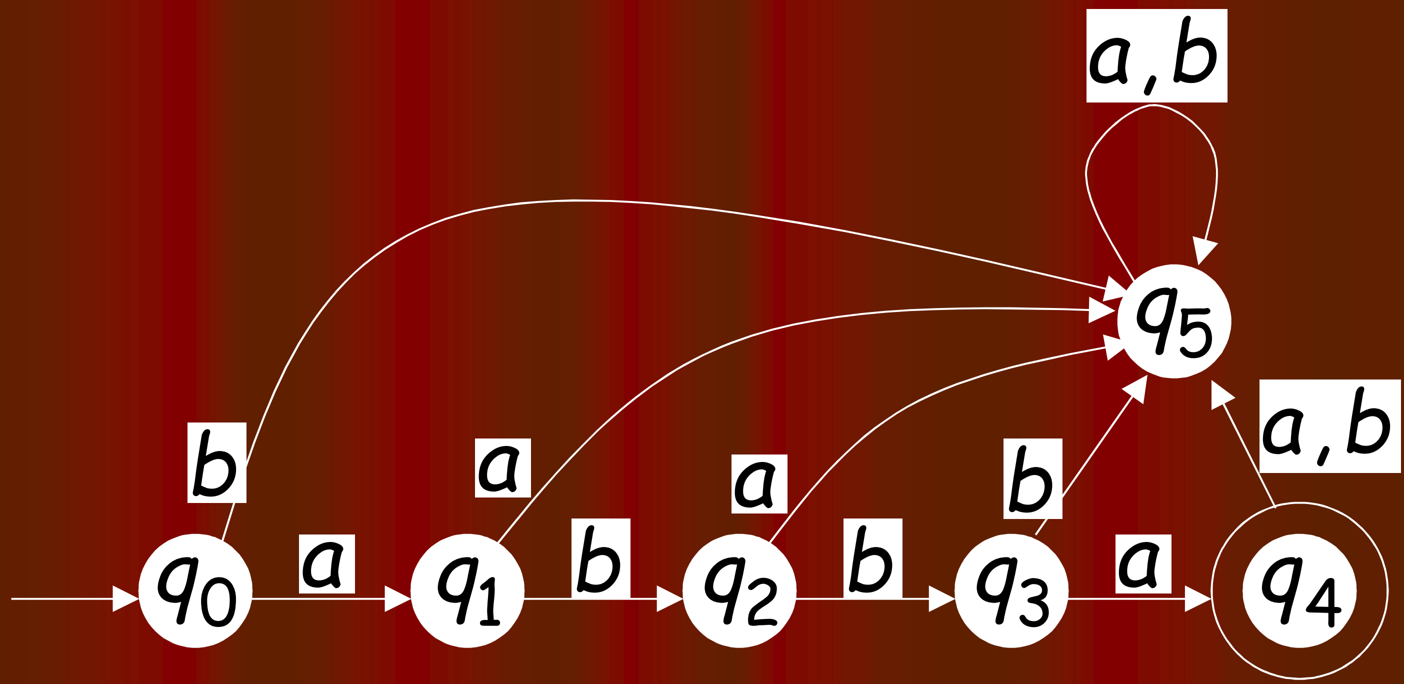
δ : transition function

q_0 : initial state

F : set of final states

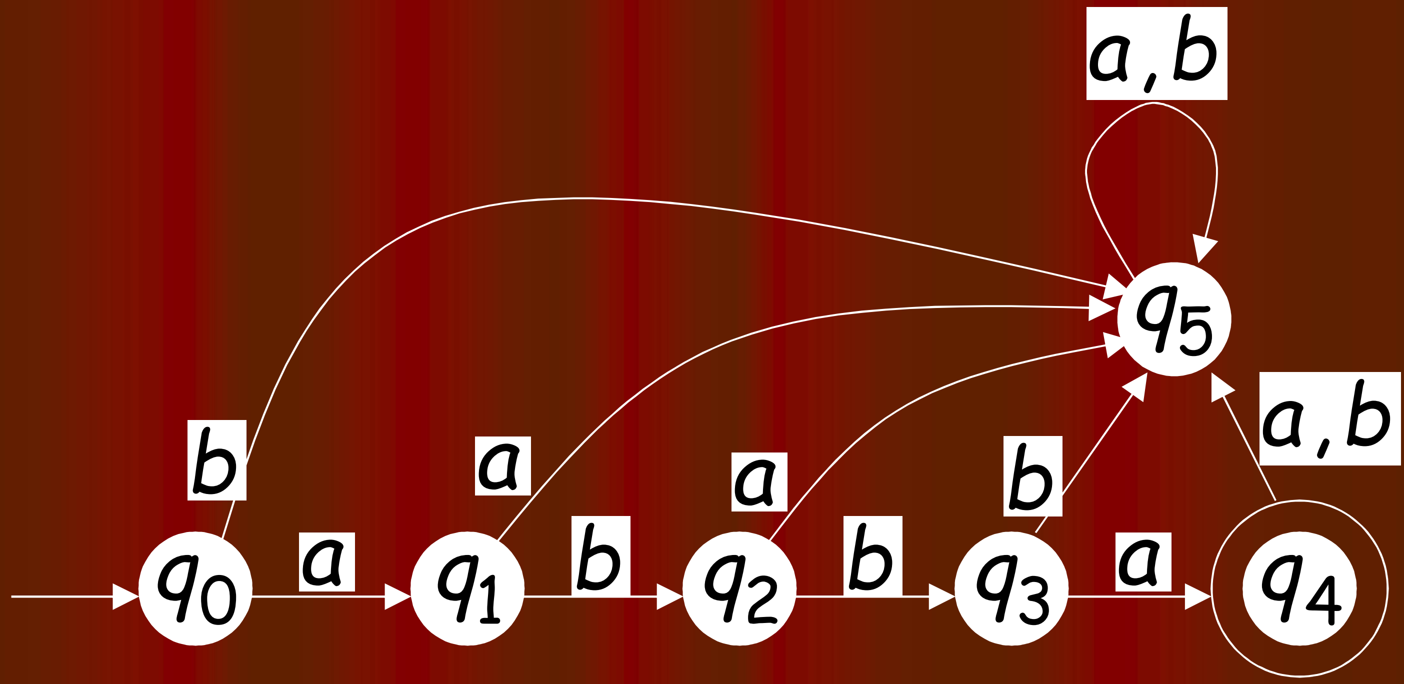
Input Alphabet

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

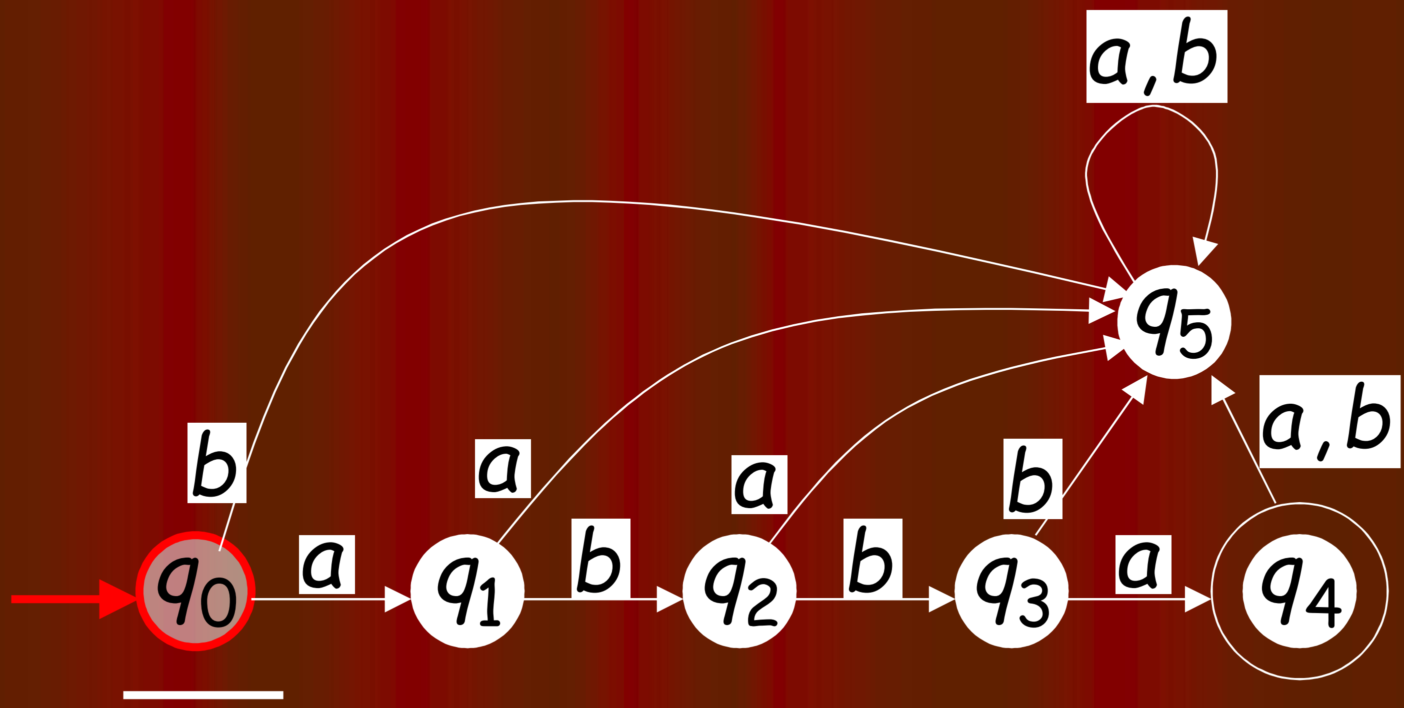


Set of States Q

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

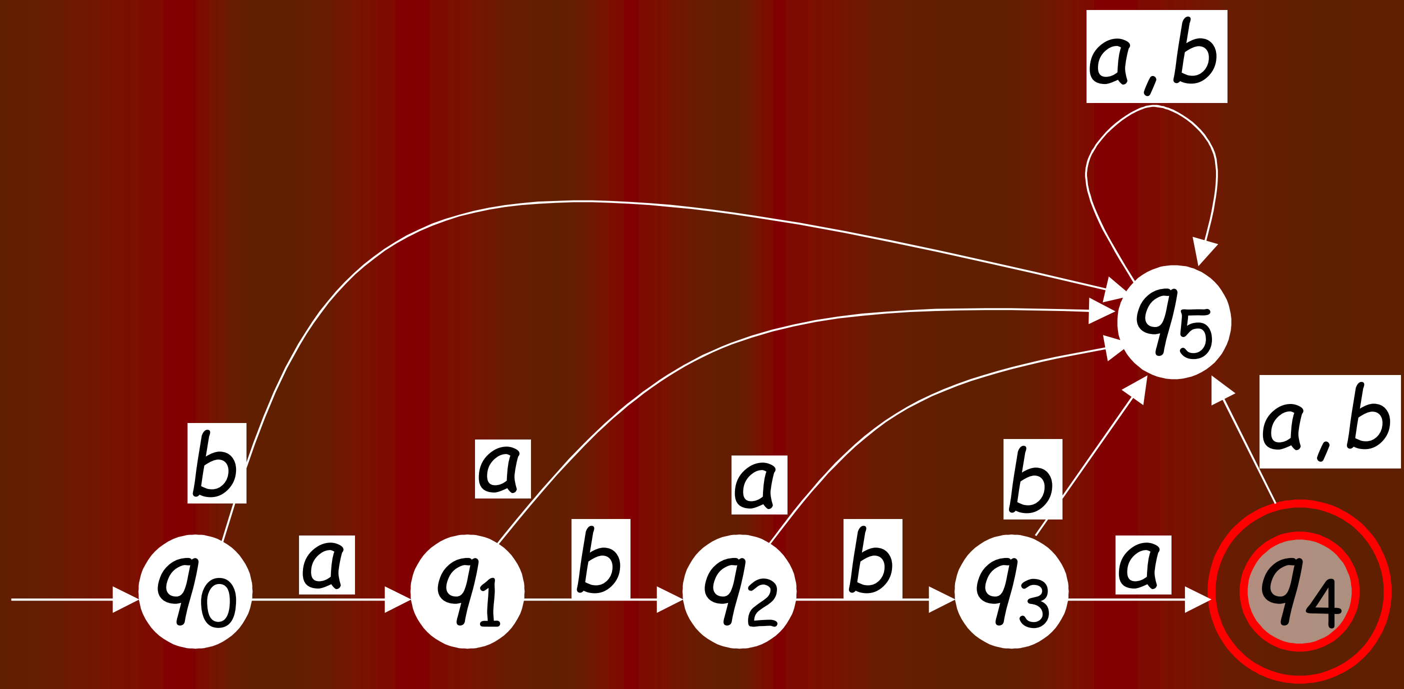


Initial State q_0



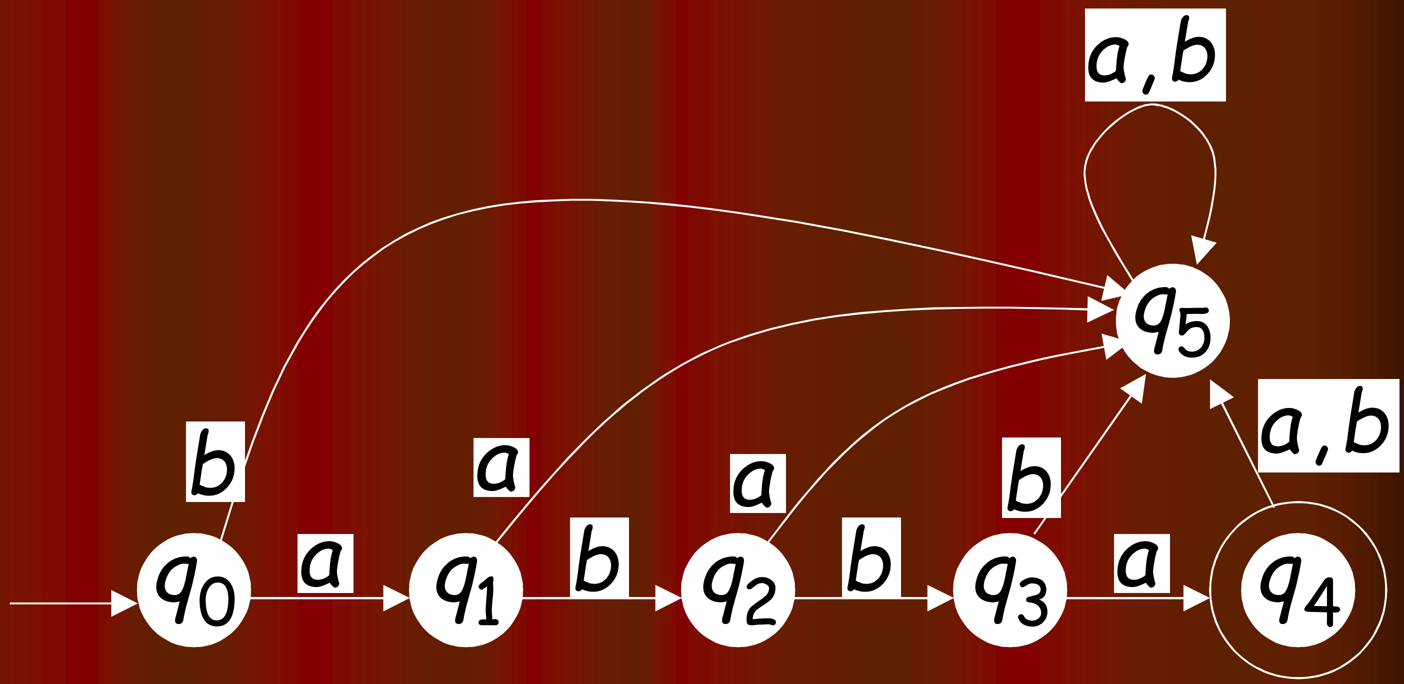
Set of Final States F

- $F = \{q_4\}$

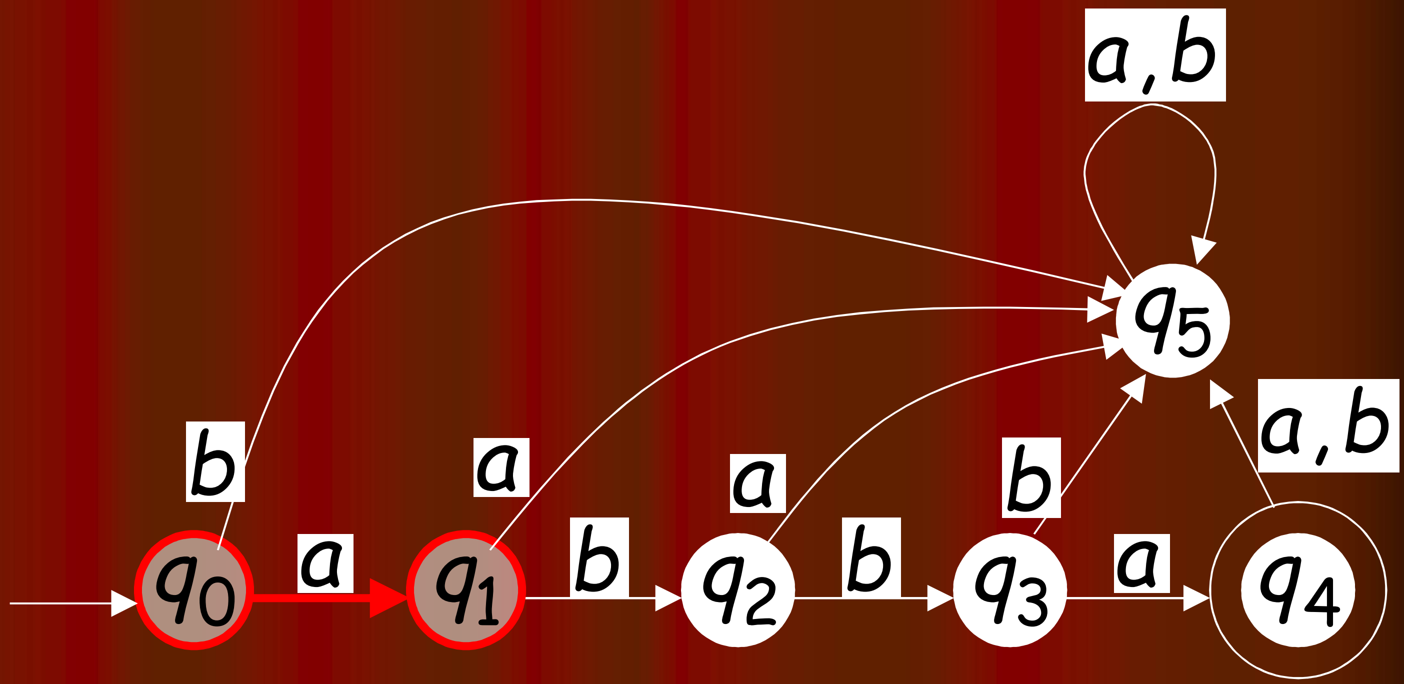


Transition Function δ

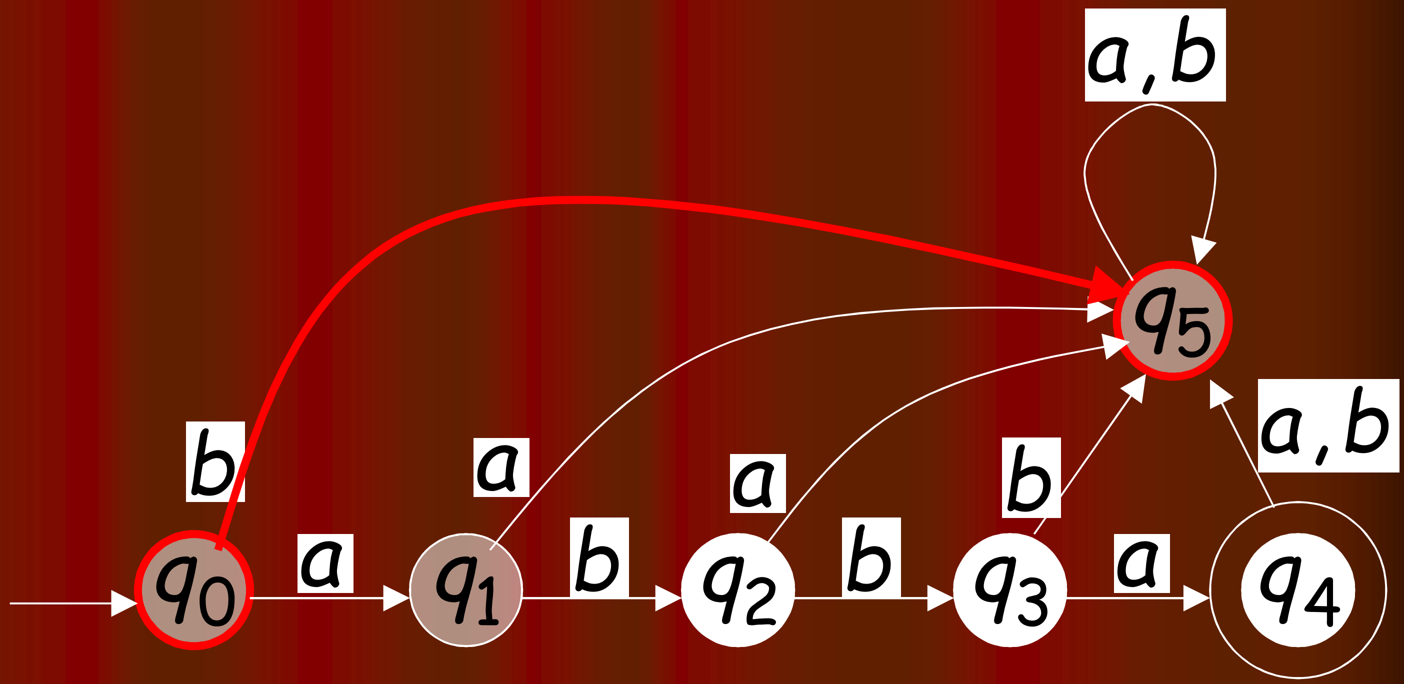
$$\delta : Q \times \Sigma \rightarrow Q$$



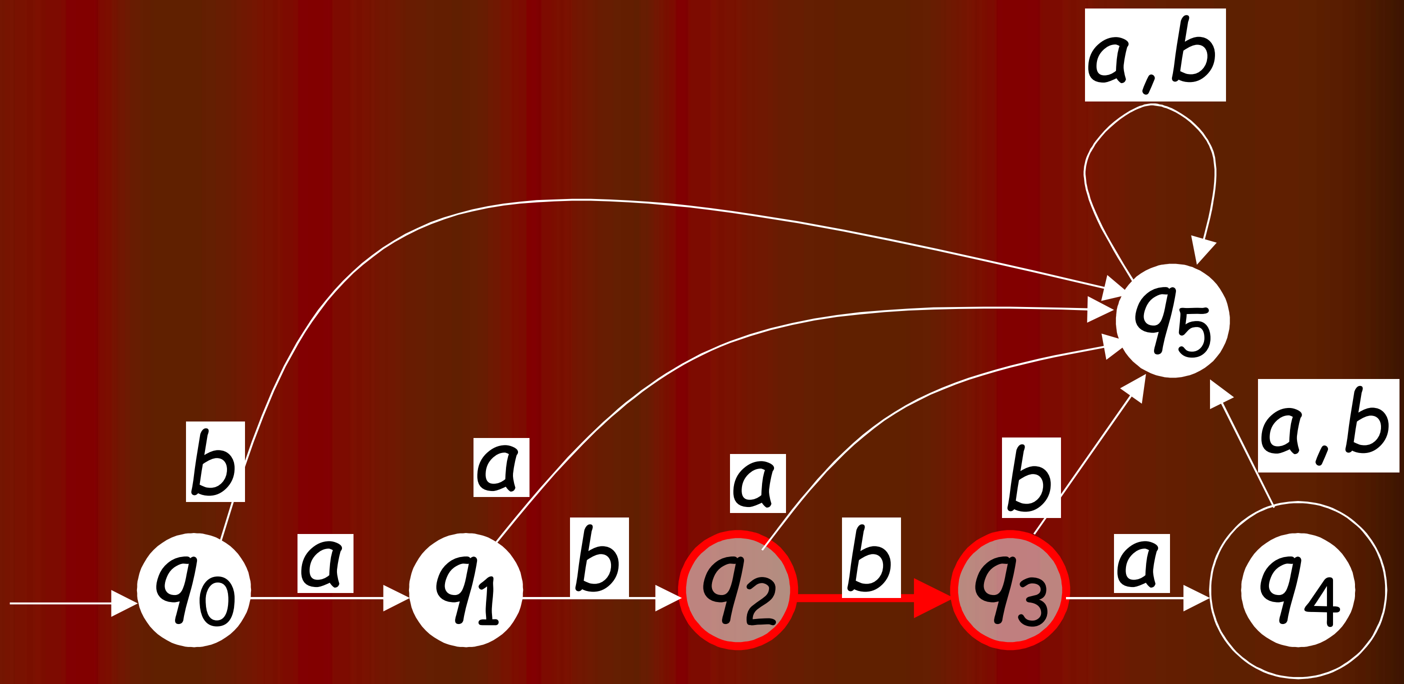
$$\delta(q_0, a) = q_1$$



$$\delta(q_0, b) = q_5$$

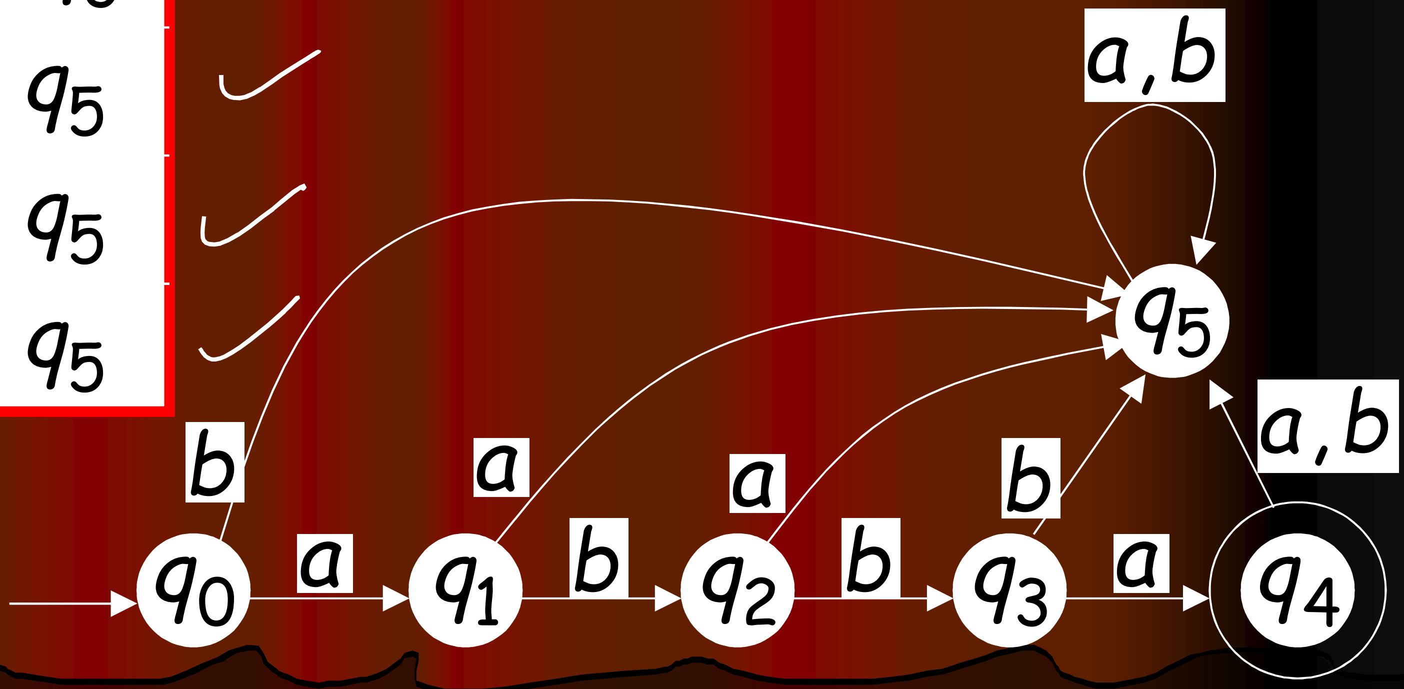


$$\delta(q_2, b) = q_3$$



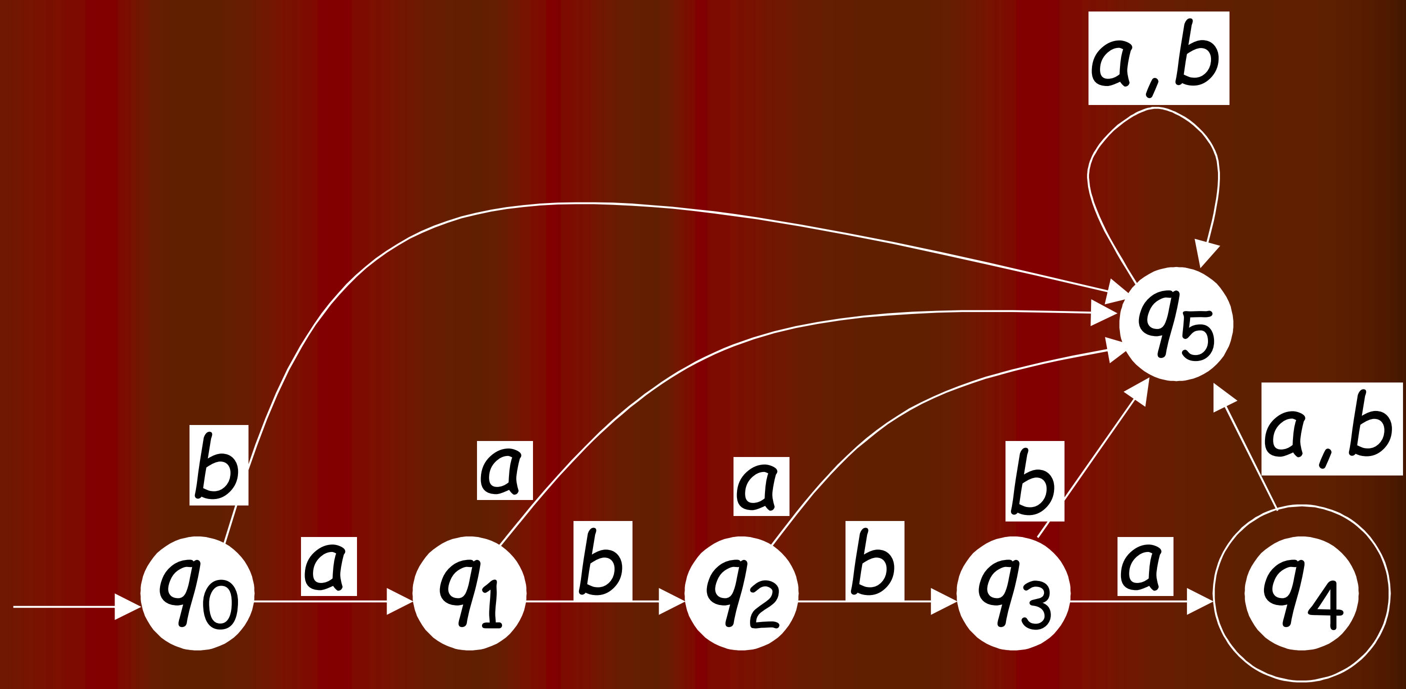
Transition Function δ

δ	a	b
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_5	q_3
q_3	q_4	q_5
q_4	q_5	q_5
q_5	q_5	q_5

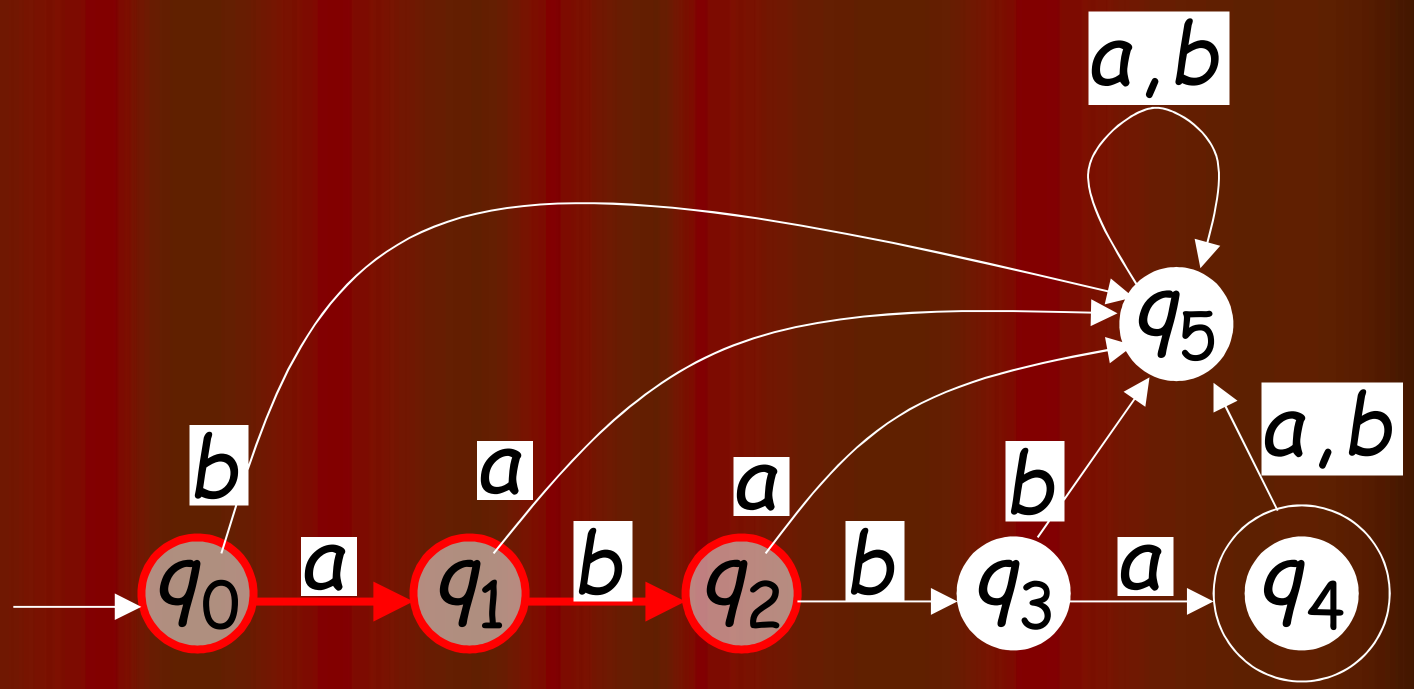


Extended Transition Function

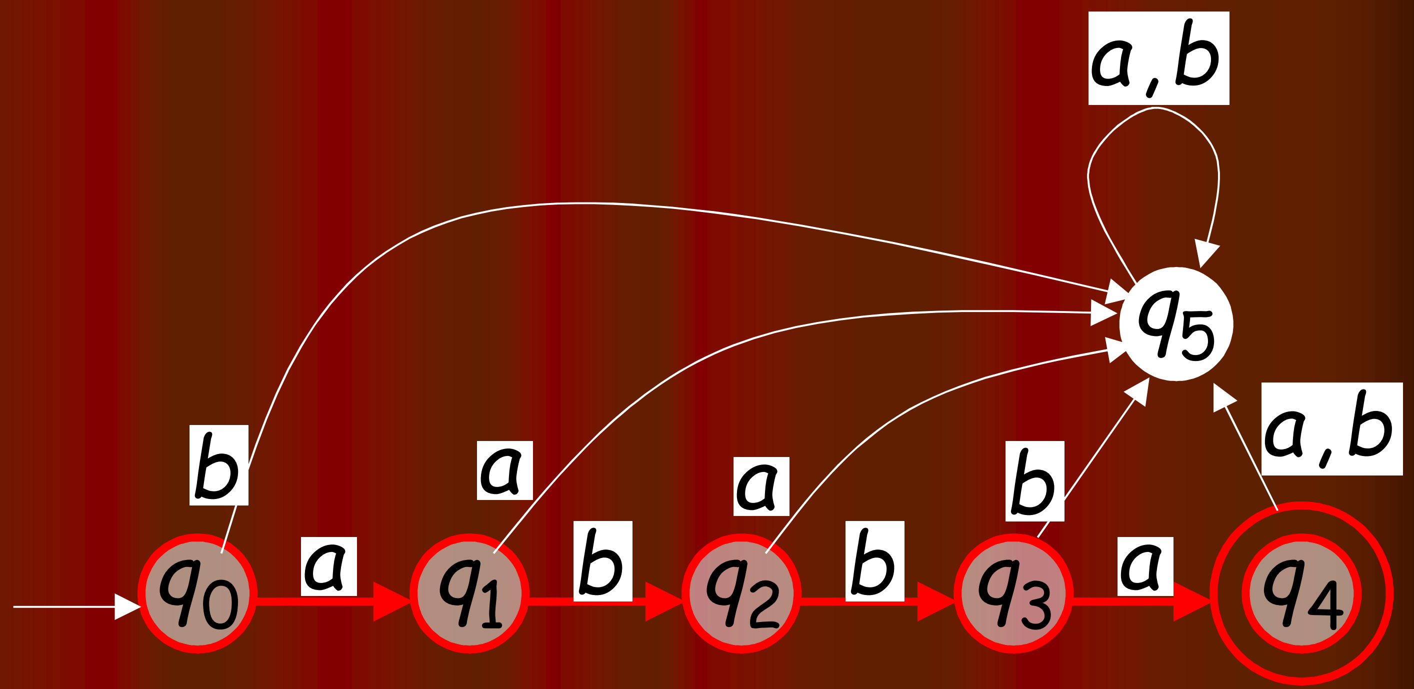
$$\delta^* : Q \times \Sigma^* \to Q$$



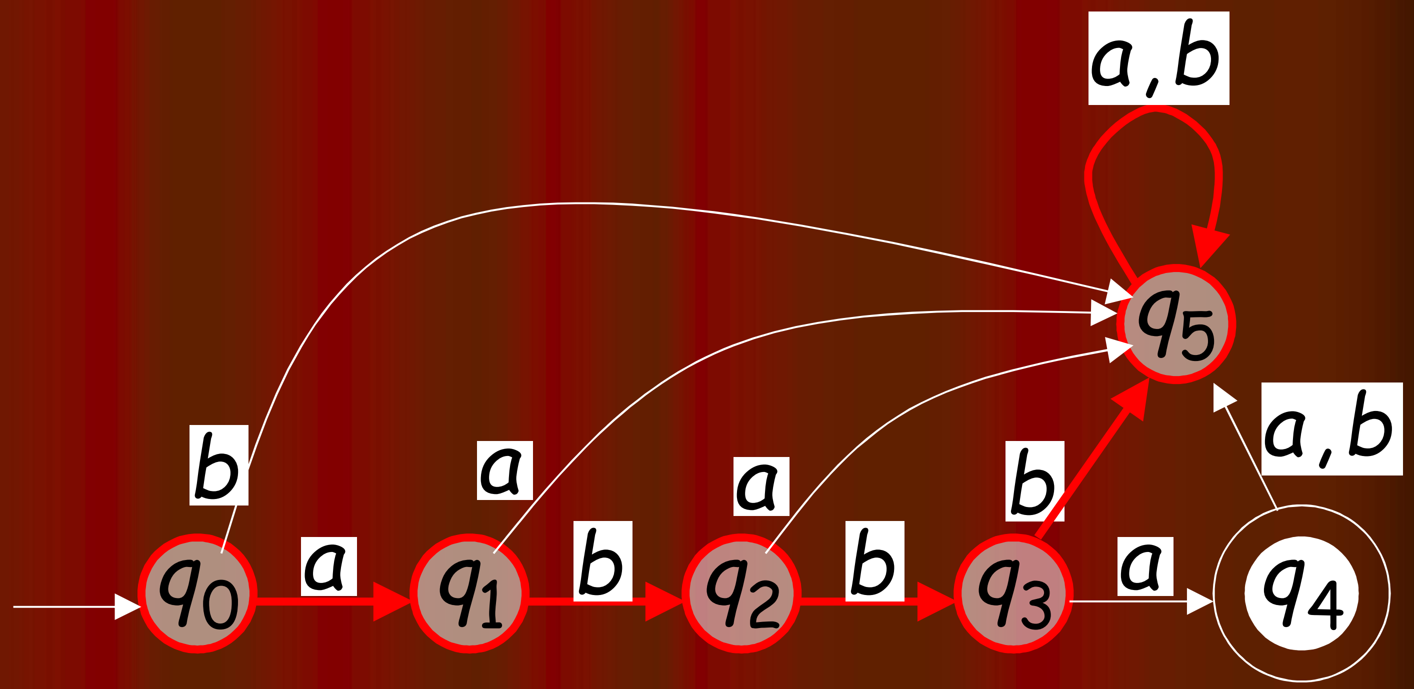
$$\delta^*(q_0, ab) = q_2$$



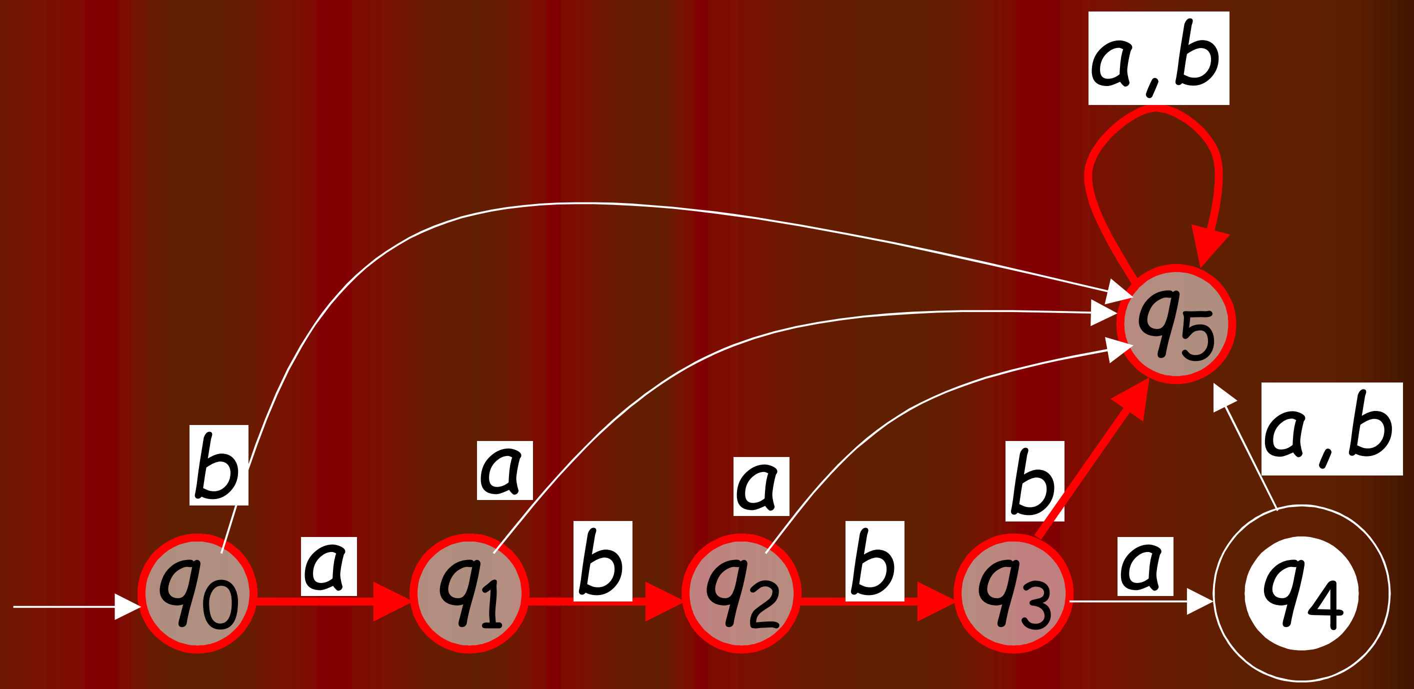
$$\delta^*(q_0, abba) = q_4$$



$$\delta^*(q_0, abbbaa) = q_5$$

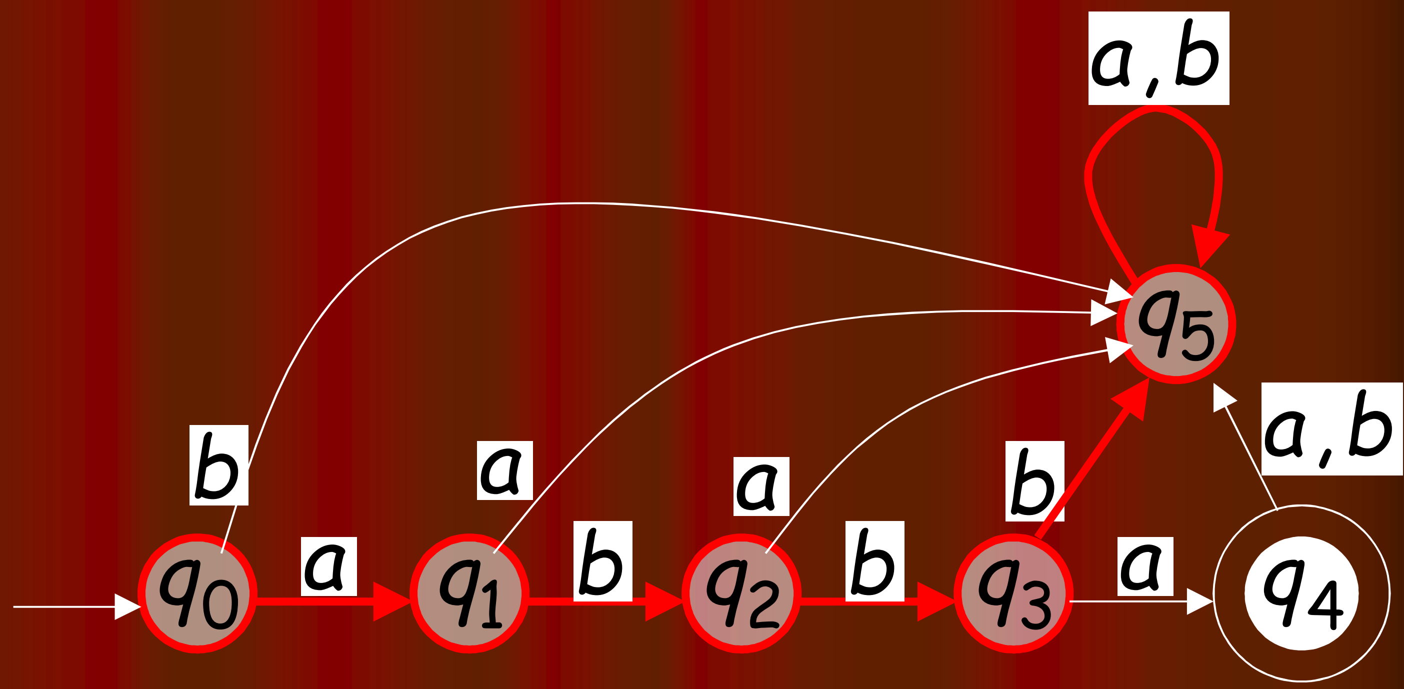


Observation: $\delta^*(q_0, abbbaa) = q_5$



Recursive Definition

- $\delta^*(q,\lambda) = q$
- $\delta^*(q,wa) = \delta(\delta^*(q,w),a)$



$$\delta^*(q_0, ab) =$$

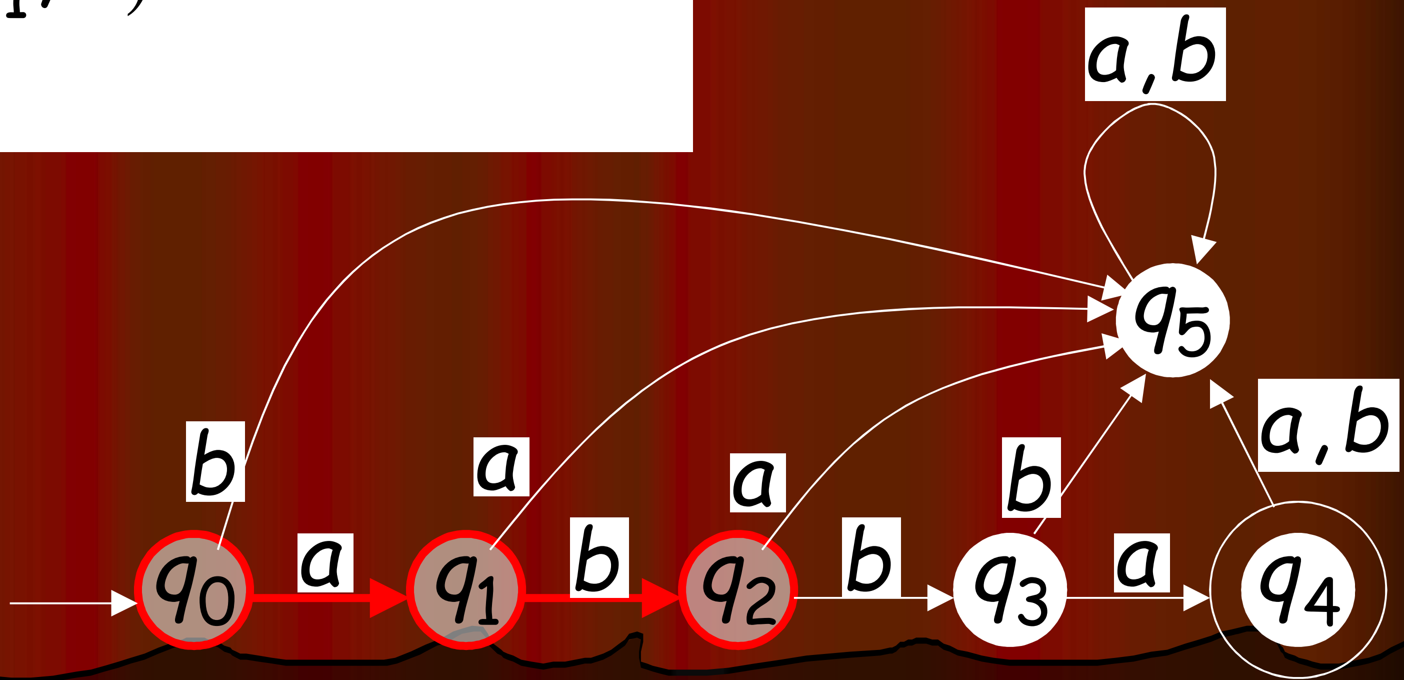
$$\delta(\delta^*(q_0, a), b) =$$

$$\delta(\delta(\delta^*(q_0, \lambda), a), b) =$$

$$\delta(\delta(q_0, a), b) =$$

$$\delta(q_1, b) =$$

$$q_2$$



Languages Accepted by DFAs

- Take DFA M

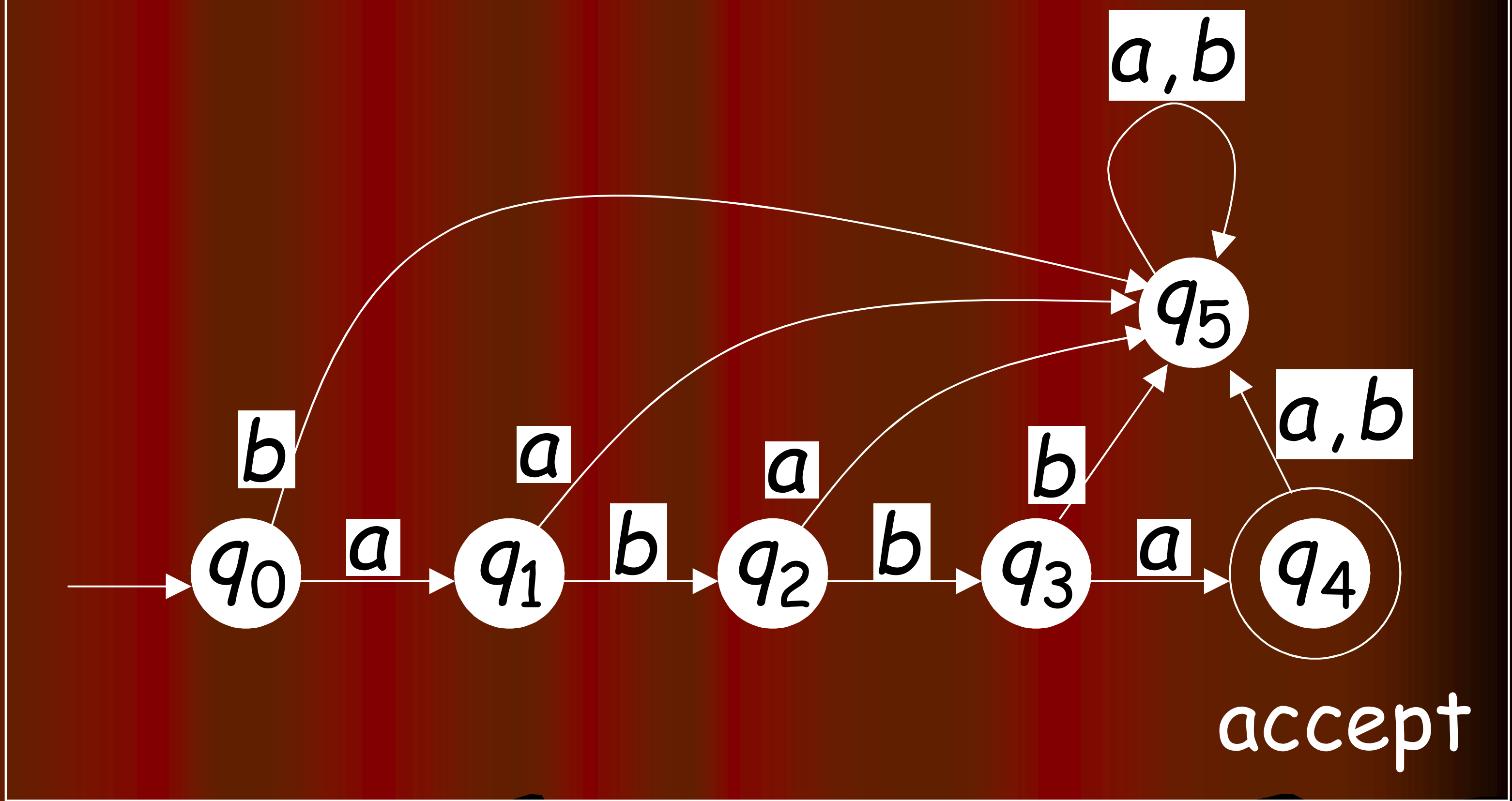
- Definition:
  - The language L(M) contains
  - all input strings accepted by M

  - L(M)= { strings that drive M to a final state}

Example

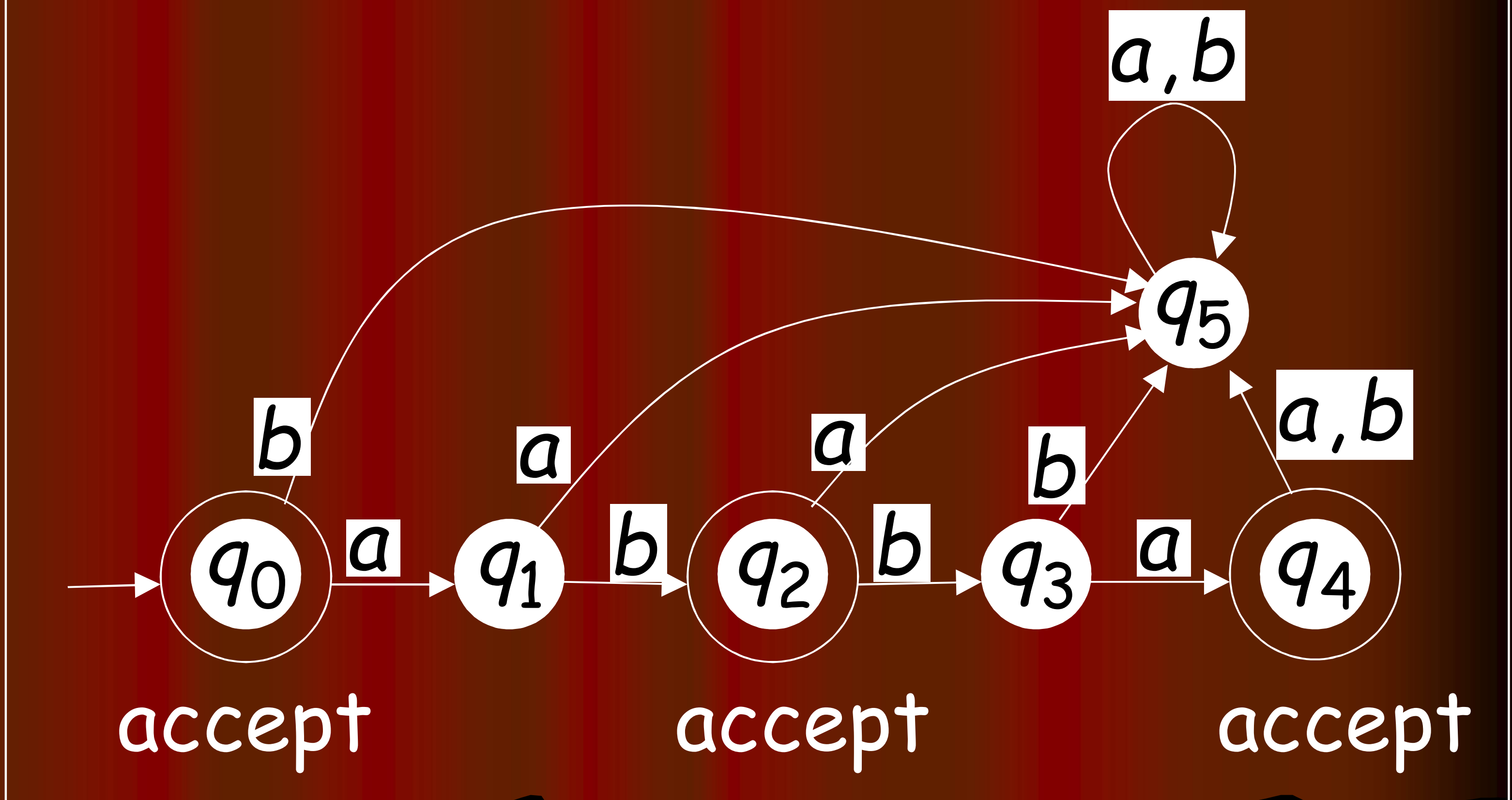
$L(M) = \{abba\}$

M



Another Example

$L(M) = \{\lambda, ab, abba\}$



Formally

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

- For a DFA
- Language accepted by M :

$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$$

alphabet

transition function

initial state

final states

Observation

- Language accepted by M :

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$$

- Language rejected by M :

$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}$$

More Examples

$$L(M) = \{a^n b : n \geq 0\}$$

$L = L(M)$